מערכות מידע ממוחשבות

ישראל בורוביץ'
זאב נוימן
יוסף מלמד

מערכות מידע ממוחשבות

מערכות מידע ממוחשבות

ישראל בורוביץ'
פרופסור חבר

זאב נוימן
פרופסור

יוסף מלמד

הפאקולטה לניהול
בית ספר למוסמכים במינהל עסקים
ע"ש לאון רקנאטי
תוכנית מחשבים ומערכות מידע
אוניברסיטת תל אביב

גומא ספרי מדע ומחקר
צ'ריקובר מוציאים לאור בע"מ

עטיפה: שרה ברקאי

כתובת ההוצאה: רחוב השרון 12, תל־אביב, 66185
דפוס ״מבט״, ת״א

תוכן הענינים

הקדמה

המחשב ומערכות המידע הממוחשבות משפיעים כיום, ואל נכון ישפיעו גם בעתיד, על כל אורחות החיים. ההכרה בעובדה זו מוצאת ביטוי במידה רבה במערכות החינוך והניהול בישראל. רוב הפאקולטות במוסדות להשכלה גבוהה כוללות בין דרישותיהן שיעורי חובה בנושא המחשבים, וניצנים ראשונים של הוראה במקצוע זה מופיעים גם בבתי־ספר תיכוניים. תכניות לקידום מנהלים והשתלמויות בתחומי הניהול כוללות במידה גיברת והולכת חלק גדול מהנושאים שבהם עוסק ספר זה.

הספר מערכות מידע ממוחשבות בא לספק את הרקע העיוני להוראת נושא המחשב ולישומו בארגונים. מטרת הספר להקנות לקורא בסיס להבנת הפעולה, המבנה והשימושים של המחשבים ומערכות המידע וכן להעמיד עבורו טרמינולוגיה שתאפשר לו תקשורת טובה יותר עם מומחי המחשבים. מטרת הספר אינה ליצור מקוראיו מומחים למחשבים, מנתחי מערכות או תוכניתנים.

הספר מערכות מידע ממוחשבות, המבוסס על ניסיון ההוראה של המחברים בנושא זה בקורסים בסיסיים לתלמידי מינהל עסקים ולתלמידי התוכניות השונות ולקידום מנהלים, נועד לשמש ספר לימוד בנושא המחשב ומערכות המידע. היקף הספר נראה למחברים כמתאים לקורס סמסטריאלי העוסק בהוראת יסודות המחשב, התכנות ומערכות מידע. כמו כן נועד הספר למנהלים ולאנשי ארגונים החשופים יותר ויותר לטכנולוגיות המידע המודרניות.

הספר מערכות מידע ממוחשבות מחולק לפרקים העוסקים בתיאור המחשב ורכיביו, בתכנות, בטכנולוגיות מידע ובהיבטים ניהוליים וכלכליים של מערכות ממוחשבות. לאחר מבוא קצר על מערכות מידע נסקרים הנושאים של חומרה ותוכנה. המחברים בדעה שכל המשתמש במערכת מידע חייב להבין מה פירוש "לתכנת מחשב". מטעם זה נכלל בפרק התכנות גם לימוד של שפת התכנות — BASIC. שפה זו נבחרה על שום היותה פשוטה, מצומצמת מבחינת כללי התחביר שלה, קלה מאוד ללימוד ונוחה לשימוש במסופים. השימוש במסופים על־ידי סטודנטים, מדענים, חוקרים ומנהלים עתיד להתפשט, ומכאן החשיבות לידיעת שפה זו, שכיום הנה שפת התכנות הנפוצה ביותר. קורא, אשר יש לו גישה למסוף המחובר למערכת המאפשרת שימוש בשפה זו, יוכל לתרגל בעצמו כתיבת תכניות. קורא שאינו מעוניין בלימוד שפה זו, יוכל לפסוח על פרק התכנות מבלי שתיפגע יכולתו להבין את שאר פרקי הספר.

מערכות ממוחשבות מודרניות מושתתות על טכנולוגיות מסדי הנתונים והתקשורת, לפיכך דן הספר גם במושגים בסיסיים הנוגעים לארגון קבצים, למסדי־נתונים ולתקשורת באמצעות רשתות מסופים ורשתות מחשבים.

מערכות מידע ממוחשבות תהוונה חלק מרכזי בכל ארגון; לכן כלול בספר פרק הדן ביישום מערכות אלו בארגון — תהליך הפיתוח של מערכות מידע, התקנתן בארגון ותהליך בחירת מערכת ממוחשבת (הכוללת תוכנה וחומרה).

המחשב, בצירוף ציוד העזר, מערכות התוכנה המפעילות אותו וכוח האדם הקשור בו, הוא משאב יקר שיש לנהלו ביעילות, כשם שמנוהל כל משאב בארגון (כספים או כוח־אדם). לפיכך דן הספר בפרק מיוחד בהיבטים ניהוליים וארגוניים של מערכות מידע ממוחשבות — ארגון ומבנה יחידת מערכות מידע בארגון, סוגי מערכות מידע והתאמתם לארגונים שונים.

בסוף הספר פרק הדן במבנה ענף המחשבים ובמגמות עתידיות בענף.

לנוחיות הקורא נרשם לצד כל מונח מקצועי המופיע בספר בעברית גם תרגומו האנגלי.

ה מ ח ב ר י ם

חלק ראשון

מבוא

פרק 1.

מערכת מידע-הגדרה, תכונות ומבנה

מבוא

מערכות מידע ממוחשבות נועדו לסייע לארגונים בתפעולם ובניהולם. מערכות מידע בארגונים מספקות מידע למקבלי החלטות ברמות שונות של ניהול.

מערכות מידע כוללות מערכות לעיבוד נתונים (Data Processing Systems), מערכות לעיבוד תנועות (Transaction Processing Systems), מערכות מידע ניהולי (Management Information Systems) ומערכות תומכות החלטה (Decision Support Systems). הדיון בכל אחת ממערכות אלו מצדיק ספר נפרד; פרק זה מקנה אפוא ידע כללי בנושא, ומספק מושג בסיסי ללומד על מערכות אלו.

המידע המיוצר ע"י מערכות מידע הנו משאב החיוני לתפעולם של הארגונים ולניהולם. ארגונים משתמשים במשאבים אחרים, כמו אנרגיה, כוח־אדם, הון, חומרים וציוד; כמויות גדלות והולכות של משאבים אלה מוקדשות למערכות מידע. שלא כמו משאבים אלה, אין המידע משאב שחייבים לרוכשו מחוץ לארגון. מערכות מידע מפותחות, מנוהלות ומנוצלות בצורה נאותה, יכולות להיות המשאב החשוב ביותר לארגון.

כדי להגדיר "מערכת מידע ארגונית" נחוץ לברר קודם שלושה מושגים — מידע, מערכת וארגון.

מידע (Information) — אלה הם הנתונים שעובדו והם בעלי משמעות למשתמש.

מערכת (System) — היא קבוצת רכיבים הפועלים ביחד כדי להשיג מטרה משותפת.
בעזרת שתי הגדרות בסיסיות אלו ניתן להגדיר מערכות מידע ארגוניות. כיוון שהמידע הוא נתונים שעובדו, הרי שנתונים מסוימים חייבים היו להיאסף, להיעבר, להיעבד ולבסוף להיאחסן. כדי שהמידע יהיה בעל משמעות עבור המשתמשים, חייבים לשלוף אותו ולהפיצו בין המשתמשים.
משתמשים אלה שייכים למערכת הנקראת "ארגון". אחד הרכיבים (או תת־מערכות) של ארגון נקרא "מערכות מידע". רכיבי מערכת זו הם אנשים, חומרה, תוכנה, נתונים ונוהלים. מערכת מידע ארגונית היא אפוא מערכת אשר אוספת, מעבירה, מעבדת ומאחסנת נתונים מחד, ומפיצה מידע למשתמשים שונים בארגון — מאידך. במלים אחרות, מערכות מידע ארגוניות מפיקות מידע שמסייע לארגון בתפעולו ובניהולו, באמצעות שימוש בחומרה ובתוכנה של מחשב, בכוח־אדם, בנהלים ידניים, במאגרי נתונים ובמודלים של החלטה וניהול.
הפיתוח, התפעול, הניהול והשימוש במערכות מידע בארגונים הנם פעילות מורכבת. מורכבות זו הולכת וגדלה בד בבד עם התקנתן של מערכות אלו בארגונים שונים, גדולים יותר ומורכבים יותר בהיקף פעילויותיהם. זו הסיבה לכך שהמחשב וטכנולוגיות המידע הקשורות אליו מהווים מרכיבים חיוניים במערכות מידע מודרניות. טכנולוגיות אלו עשויות להפוך לקללת מערכות המידע במקום לברכה, אם לא יעוצבו בצורה נאותה במערכות ויבוקרו אחר כך.
ראוי לציין שדיוננו נוגע למערכות מידע ארגוניות ולא למערכת מידע כוללת אחת. בארגונים קיים אגד של מערכות מידע שמקל על תפעול הארגון וניהולו ומספק מידע לצורכי בקרה ניהולית. מערכות המידע השונות באגד (ניהול מלאי, הנהלת חשבונות וכו') יכולות להיות קשורות ביניהן בצורה חזקה או חלשה; הן יכולות להשתמש במשאבים משותפים או להיות בעלות משאבים עצמאיים. מה שמאחד את כל המערכות הללו הוא התכנון והבקרה הנגזרים ממטרות הארגון, המבנה והאסטרטגיה שלו על־ידי ההנהלה הבכירה של הארגון.

כדי להבין את מהותן, את תפקידן ואת מבניהן של מערכות מידע ארגוניות יש לברר את הנושאים האלה:

- גישת המערכות
- מערכת הארגון
- מערכת הניהול
- מבנה של מערכת מידע
- גישות לבניית מערכות מידע.

שלושת הנושאים הראשונים ידונו בהמשך. שני הנושאים האחרונים ידונו בפרק מס' 5.

גישת המערכות

גישת המערכות עיקרה ראיית השלם המורכב מבלי לאבד בו־זמנית את ראיית רכיביו של אותו שלם. לשון אחר: היכולת לטפל ברכיב (תת־מערכת) תוך ראיית אותו רכיב כחלק מהמכלול שהוא משתייך אליו (מערכת).
גישת המערכות היא אוניברסלית מחד, היינו דרך מחשבה ופילוסופיה של ראיית דברים, ומאידך משמשת כלי לניתוח בעיות, מצבים וגופים מורכבים.

מהי מערכת?
מערכת היא קבוצת גורמים הקשורים ביניהם, והיוצרים שלמות מורכבת אחת במטרה להשיג יעד מוגדר. קבוצת הגורמים יוצרת מערכת רק כאשר הגורמים הבונים אותה קשורים זה לזה וקיימת ביניהם תלות ופעילות הדדית. מערכת תהיה מורכבת אפוא משני רכיבים לפחות, בעלי יחס גומלין (אינטראקציה) ביניהם. אוסף רכיבים שאינו מקיים פעילות גומלין אינו מערכת. היות שהמערכות שאנו עוסקים בהן הן מערכות העשויות בידי אדם (בניגוד למערכות טבעיות), הרי שיש להן מטרה (או מטרות). רכיבי המערכת יכולים להיות הם־עצמם מערכות ואז הם יהוו תת־מערכת (sub-system), ואילו המערכת הנידונה יכולה להיות רכיב במערכת־על (super system or metasystem). אותה דרך ניתוח טובה לתת־מערכת, מערכת ומערכת־על.
דרך נוחה ללימוד ולניתוח מערכת היא קביעת תכונותיהם של המשתנים העיקריים המאפיינים את המערכת.
להלן פירוט המשתנים העיקריים הללו (ראה תרשים 1.1):

א. גבולות המערכת

גבול מערכת הוא תחום פיסי או לוגי המפריד את המערכת הן מסביבתה והן ממערכות אחרות.
לדוגמה: הגבול של מערכת הטלפון המקומית הוא כל הטלפונים המחוברים למרכזת מסוימת אחת; גבול מערכת חברתית (משפחה למשל) הוא קבוצת אילוצים המונעים חלק מהמידע הפנימי שלה להיות זמין לסביבתה, כולל למערכת העל שלה (לשבט למשל).
בדרך־כלל יש מעבר של משאבים ומידע פחות אל מעבר לגבולות המערכת מאשר בתוך המערכת.
שתי הדוגמאות ממחישות את ההבדל בין גבולות פיסיים לבין גבולות מושגיים־לוגיים.
הגבולות תוחמים את המערכת. קביעת הגבולות אינה פעולה פשוטה וישירה בכל מקרה. פעמים התהליך הוא איטרטיבי (חוזר אחורה לשלבים מוקדמים), ויש בו מידה רבה של שרירותיות. התיחום מתבצע על סמך המידע הזמין בעת לימוד וניתוח המערכת, הניסיון והידע שיש למבצע התיחום, ומטרת הניתוח. מטרה

נוספת של התיחום (השרירותי) היא שמירה על מערכת שהיא בת־טיפול או בת־תפיסה (מבחינת היקף, סיבוך, יכולת תפיסה אנושית, יכולת שליטה, מחיר, זמן).

המושג ״גבול של מערכת״ חשוב לתכן (design) של מערכות מידע. ההכרה בצורך התיחום כופה על המתכנים לשקול בחירה בין היקף לבין מורכבות.

ב. המטרות ומדדי הביצוע של המערכת

מערכות העשויות בידי אדם הן בדרך־כלל מכוונות למטרה (purposeful systems). הבנה והגדרה של מטרות המערכת היא תנאי ראשון להצלחת ניתוחה. אולם לא די בהבנת המטרות ובהגדרתן. יש להבין ולהגדיר גם את מדדי הביצוע, שלפיהם שופטים את טיב תיפקודה של המערכת להשגת מטרותיה. קיימים שני סוגי מדדים — מדדי הישגיות (אפקטיביות) ומדדי יעילות — העשויים לגלות באיזו מידה השיגה המערכת את מטרתה ועד כמה היתה יעילה בתהליך השגת המטרה. בעוד מדד ההישגיות עונה על השאלה: ״האם אנו עושים את הדבר הנכון?״ עונה מדד היעילות על השאלה: ״האם אנו עושים דברים בצורה הנכונה?״

כך למשל המטרה של אגף התנועה במשטרת ישראל יכולה להיות שמירה על כבישים ללא תאונות. מדד הישגיות אפשרי הוא מספר התאונות לק״מ כביש; ככל שמדד זה נמוך יותר, כך הישגי יותר אגף התנועה בהשגת מטרתו. אולם כיוון שלאגף יש משאבים מוגבלים (תקציב, מכוניות, שוטרים), אנו מעוניינים למדוד את יעילותו בניצול המשאבים. מדדים אפשריים של יעילות יהיו מספר התאונות לק״מ כביש ליום שוטר, או לשקל תקציב. ככל שמדדים אלו גבוהים יותר, כך האגף יעיל פחות בהשגת מטרתו.

ג. קשרי המערכת עם סביבתה (קלטים ופלטים)

המערכות שבהן אנו עוסקים הנן מערכות פתוחות שקשרי גומלין להן עם סביבתן: קלטים נכנסים מהסביבה אל המערכת ופלטים יוצאים ממנה אל סביבתה. קלטים ופלטים אלה, או במלים אחרות, התשומות והתפוקות, יכולים ללבוש כל צורה שהיא: חומר, אנרגיה או מידע. מערכות ארגוניות שהן מערכות חברתיות הן תמיד מערכות פתוחות.

ד. רכיבי המערכת (תתי־מערכות)

אפיון רכיבי המערכת הוא פעולה הדומה לפעולת תחימת המערכת בין גבולות. פעולה זו אינה תמיד חד־משמעית, ישירה ופשוטה, והיא יכולה להיעשות ברמות תמצות שונות. הרמה תקבע על־ידי מטרת הניתוח, טיבו של המנתח, נסיונו, נקודת מבטו, וכן הקפה ומורכבותה של המערכת, ועל כן, ויש בה מידה מסוימת של שרירותיות. אפיון כל רכיב שהוא תת־מערכת במערכת ייעשה בדרך שהותוותה למערכת עצמה.

לגבי כל רכיב במערכת יש לקבוע:

- — מהן מטרותיו ומה הקשר שלו למטרות המערכת שלה הוא שייך
- — מהם תהליכי העיבוד המתבצעים על ידו
- — מהם המשאבים שבהם הוא עושה שימוש
- — מהו מקומו במדרג רכיבי המערכת.

אם אין מנתחים את המבנה הפנימי של הרכיבים, יש להתייחס אל הרכיב כאל "תיבה שחורה". התייחסות כזו פירושה ניתוח הקלטים והפלטים והבנתם ללא לימוד המבנה הפנימי והתהליכים המבצעים את ההמרה (טרנספורמציה) של הקלט לפלט בתוך הרכיב. שיטה זו יעילה כאשר עוסקים במערכות מורכבות בעלות מספר רב של רכיבים והתעניינות, או מגבלת כוח־אדם בצוות הניתוח, מכתיבים התמקדות רק בחלק מהרכיבים. הדבר יעיל גם כאשר נמצאים בשלב מוקדם של פעולת הניתוח, ורק המבנה הכללי של המערכת הוא המטרה של הניתוח. אולם בשלבים מתקדמים יותר של הניתוח, כאשר המבנה הכללי מנותח ומוגדר, יורדים אל המבנה הפנימי של רכיבי המערכת.
יש להבחין בין שני סוגי רכיבים: אלה העוסקים בהמרה של חומר ואנרגיה ואלה העוסקים בהמרה של מידע. ראוי לציין שגם הרכיבים העוסקים בהמרה של מידע עושים שימוש בחומר ובאנרגיה (למשל, ציוד מחשב, כוח־אדם).

ה. קשרי הגומלין בין רכיבי המערכת

קשרי הגומלין בין הרכיבים הם ההופכים את קבוצת הרכיבים למערכת. קשרי גומלין אלה הם אחד הגורמים העיקריים למורכבותו של הניתוח, משום שניתוח המערכת פירושו ניתוח מאפייני המערכת כיחידה שלמה, ניתוח רכיביה וניתוח קשרי הגומלין ביניהם. רכיבי המערכת וקשרי הגומלין ביניהם נקראים מבנה המערכת. הקשר בין הרכיבים מתבצע באמצעות מימשקים (interfaces), שהם ערוצי קלט ופלט היכולים ללבוש צורה של חומר, אנרגיה או מידע.
יש לזכור שקיימים גם קשרי גומלין בין רכיבי המערכת לבין סביבתה, וגם את אלה יש להגדיר ולנתח. רשת התקשורת בתוך המערכת מהווה חלק מקשרי הגומלין בין הרכיבים, כשמדובר בקשרי גומלין של מידע.
הצוות המפתח מערכת מידע עושה שימוש בגישת המערכות, בכך שהוא קובע את מבנה המערכת תוך קביעת המבנה המידרגי, קביעת מאפייני כל רכיב וקביעת קשרי הגומלין בין הרכיבים.

ו. סביבת המערכת

סביבת המערכת היא כל שנמצא מחוץ לגבולותיה. אולם הסביבה, בהתאם להגדרה זאת, היא אינסופית. לפיכך, סביבתה של המערכת אשר אותה מנתחים,

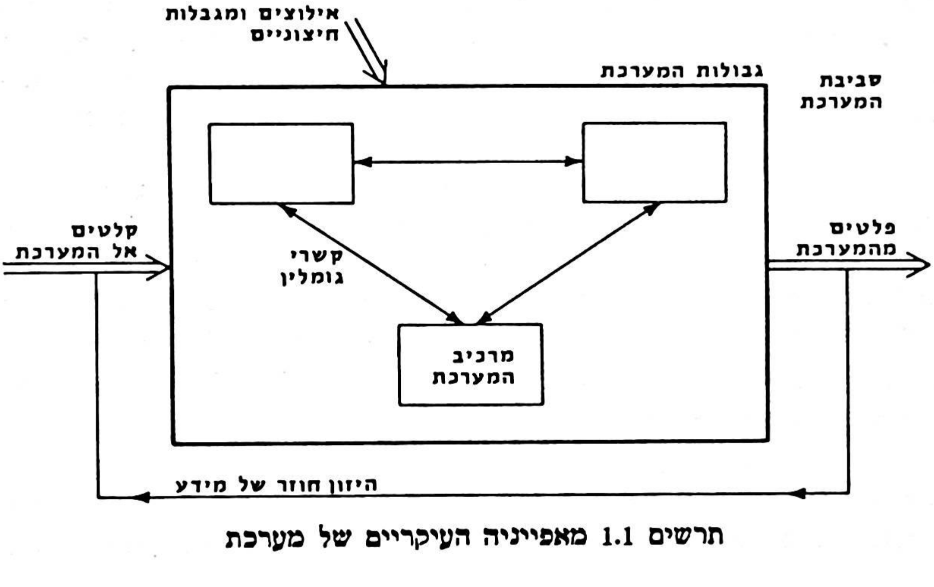
היא הסביבה הרלוונטית לה, לאמור הסביבה שעמה יש למערכת קשרי גומלין. את הסביבה הרלוונטית ניתן לחלק לסביבה קרובה ולסביבה רחוקה.

ז. מגבלות ואילוצים הפועלים על המערכת

על כל מערכת פועלים מגבלות ואילוצים אשר יש לאתרם, לזהותם ולנתחם. יש להבחין בין מגבלות ואילוצים חיצוניים, הפועלים על המערכת מתוך סביבתה הרלוונטית, לבין מגבלות ואילוצים הפועלים בתוך המערכת כפועל יוצא ממדיניותה, ממבניה ומתכונותיה.

ח. הגורם המבקר או תת־מערכת הבקרה

תת־מערכת הדרושה לכל מערכת מכוונת־מטרה היא תת־מערכת הבקרה. זהו אותו רכיב של המערכת המקבל מידע מכל גורמי המערכת ומסביבתה הרלוונטית, מקבל החלטות ומעביר הוראות שביצוען מאפשר תיפקוד יעיל, נכון ומכוון־מטרה של המערכת ושומר על היות פלט המערכת בין גבולות רצויים.
במערכת האדם — המוח הנו הגורם המבקר. בארגונים — זוהי מערכת הניהול המתפקדת על בסיס המידע הבא אליה כהיזון־חוזר (feed-back) המתאר את פעולת המערכת (היזון חוזר פנימי) ועל בסיס מידע הבא אליה כהיזון חוזר מהסביבה הרלוונטית (היזון חוזר חיצוני).
יכולתה של המערכת להגיב להפרעות ולשינויים תלויה בטיבו של המידע הבא אליה כהיזון חוזר, בעיכובים בתמסורת שלו ובתהליך הבקרה המתאים את משתני המערכת כתגובה לאותות ההיזון החוזר (המידע).



תרשים 1.1 מאפייניה העיקריים של מערכת

מערכת הארגון

הארגון הנו המערכת שעבורו בונים את מערכת המידע, משלבים אותה בתוכו ואותו היא משרתת. כדי לבנות את מערכת המידע יש לנתח תחילה את מערכת הארגון. מטרת הניתוח היא איתור צורכי המידע של הארגון ובחינת מקומה של מערכת המידע ותפקידה באספקת צרכים אלה.

הארגון הנו מערכת ממשית, חברתית, אולם הוא כולל גם רכיבים פיסיים ומופשטים. תפקידו המוגדר של הארגון הוא הפיכת משאבים (באמצעות המרה) למוצרים, לשירותים או לרעיונות שניתן לשלבם בחברה.

המשאבים העוברים המרה הם הקלט של המערכת, ואילו המוצרים, השירותים, או הרעיונות הם פלטי המערכת, שתפקידם השגת מטרות המערכת.

רכיבי מערכת הארגון

כל ארגון, ברמת מורכבות והיקף כלשהם, כולל בדרך כלל שלושה סוגי רכיבים (רכיב אחד מכל סוג לפחות):

— רכיב התפעול

תפקידו לבצע המרה של משאבים למוצרים, לשירותים או לרעיונות. זהו הרכיב העיקרי של הארגון, משום שתפקידו להשיג את מטרות הארגון.

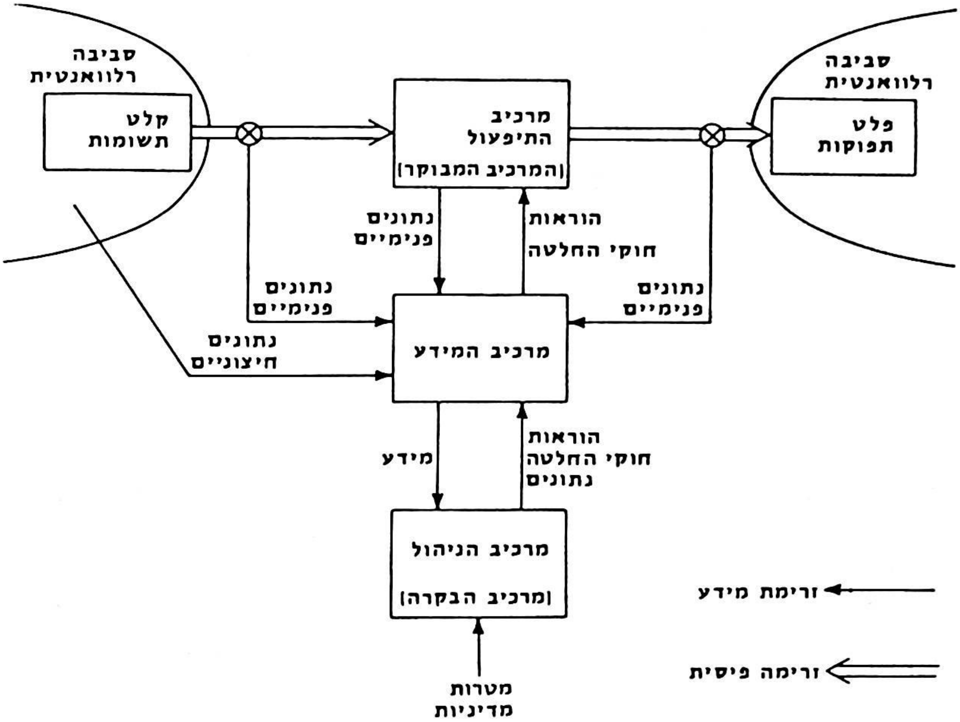
— רכיב הבקרה

תפקידו לבצע את פעולת הבקרה של רכיבי התפעול (שהם הרכיב המבוקר).

— רכיב המידע

תפקידו כפול: מחד עליו להעביר לרכיב התפעול את ההוראות שמקורן ברכיב הניהול, ומאידך למסור לרכיב הניהול מידע על המתרחש ברכיב התפעול (היזון חוזר של מידע).

רשת תקשורת מקשרת בין שלושת רכיבי הארגון (ראה תרשים 1.2)



תרשים 1.2 מודל מכוגנן ומופשט של הארגון

רכיב המידע משמש כמתווך בין רכיב התפעול לרכיב הניהול ונותן לרכיב הניהול תמונה מופשטת על המתרחש ברכיבי התפעול. קיימת אבחנה ברורה בין הזרימה הפיסית המתרחשת בין רכיבי התפעול לבין זרימת המידע במערכת המידע.

ניתוח מערכת הארגון

ניתוח מערכת הארגון מצריך ביצוע סקר שיטתי שייתן תמונה מפורטת למדי של הארגון שאותו מנתחים. כך ניתן לאתר את צרכי המידע של הארגון, את נמעני המידע, את מיקומה של מערכת המידע בארגון וכן את מדיניות הקמתה של מערכת המידע.

בביצוע הסקר יש לחפש תשובות לשאלות האלה:

א. מהן מטרות הארגון ומהם מדדי הביצוע שלו בהשגת המטרות
ב. מהם גבולות הארגון
ג. מהם משאבי הארגון — זרמי המשאבים הנכנסים אליו ומאגרי המשאבים המצויים בו
ד. מהם הגורמים הבונים את הארגון ומהם התהליכים המתבצעים בהם (את התהליכים יש לסווג לתהליכים תפעוליים, נהוליים ותהליכי עיבוד מידע).
ה. מהם קשרי הגומלין בין הגורמים
ו. מהי התייחסותו של כל גורם במערכת למטרות הארגון
ז. מהי הסביבה הרלוונטית של הארגון (הקרובה והרחוקה), מהם קשרי הגומלין של המערכת עמה וכן מהם האילוצים והמגבלות שסביבה זו מפעילה על המערכת
ח. מהם דרכי הפעולה העומדות לרשות הארגון והמאפשרות לו להשיג את מטרותיו, וכן מהי יכולתו ליצור דרכי פעולה חדשות בעת הצורך
ט. מהם צרכי המידע של כל גורם בארגון
י. מהם מקורות המידע של הארגון (פנימיים וחיצוניים).

מערכת הניהול

רכיב הניהול בארגון הנו תת־מערכת של הארגון. הוא הצרכן העיקרי של מערכת המידע, משום שהמידע שהיא מפיקה הוא הכלי העיקרי המסייע לניהול. לפיכך, בנייתה של מערכת המידע תתבצע נכונה רק אם נבין את מהותה וצרכיה של מערכת הניהול שאותה היא משרתת.
תהליך הניהול הנו תהליך של המרה שבו הופך מידע (הקלט) להחלטות (הפלט). ראשיתו של התהליך הנ"ל בבקרה: על תהליך הבקרה המופעל על רכיב התפעול מוטלת האחריות להפוך את מערכת המשאבים — הרכיב התפעולי (כוח אדם, חומרים, טכנולוגיה, אנרגיה, הון, אמצעים פיסיים ומידע) — למערכת הרמונית (שילוב, איחוד וקישור בין רכיביה של מערכת המשאבים) המסוגלת להשיג ביעילות (מוכתבת על־ידי מדדי הביצוע) את המטרות שהוצבו בפניה (הפקת הפלט שהוא: מוצרים, שירותים או רעיונות).
עקרונית זהה תהליך הניהול בכל דרגי הניהול של הארגון, ממנהל העבודה ועד המנכ"ל. התהליך הנו שילוב של אמנות עם בסיס מדעי ומשום היותו כזה תלותו באדם המבצע אותו מכרעת.

הפונקציות המרכיבות את תהליך הניהול

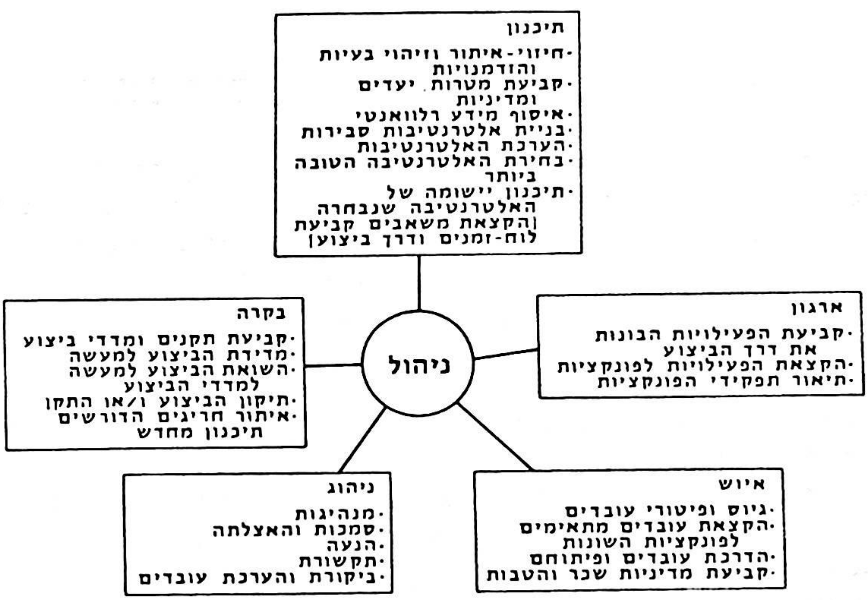
כדי להגיע לדרך הניתוח המתאימה לצרכינו עלינו לפרק את תהליך הניהול לפונקציות שממנו הוא מורכב. אין מדובר כאן בפונקציות מקצועיות, כמו שיווק, ייצור, כספים וכוח־אדם, אלא בפונקציות הניהול האוניברסליות אשר מבצעים מנהלים בכל הדרגים. משקלן הסגולי של הפונקציות משתנה מדרג ניהול אחד למשנהו (מבחינת הדגשים, היקף חשיבות, כמות הזמן המושקעת וכו׳).

הפונקציות שמהן בנוי תהליך הניהול: (תרשים 1.3)

— תכנון (planning)
— ארגון (organizing)
— איוש (staffing)
— ניהוג (directing)
— בקרה (controlling)

בכל אחת מפונקציות הניהול קיימת פעילות ניהולית מסוימת:

— קבלת החלטות
— קשרי־גומלין בין־אישיים
— תיאום ותקשורת
— מיבנות (structuring) פעילויות.



תרשים 1.3 תהליך הניהול

פירוק תהליך הניהול לפונקציות המרכיבות אותו הוא פעולה מתודולוגית במהותה, לאמור פירוקו של התהליך לחמשת הפונקציות הללו הוא מעשה שרירותי במידה רבה ותכליתו ניתוח ולימוד תהליך הניהול. הגבולות בין הפונקציות אינם חד־משמעיים וביצועו של התהליך במציאות אינו מתבצע בסדר מוכתב מראש דווקא, תוך הפרדה בין הפונקציות השונות. אבל התועלת שבהפרדת התהליך לפונקציות היא באיתור צורכי המידע של מערכת הניהול. ואמנם, כל פונקציה ניהולית וצרכיה היחודיים, כל פונקציה — ויכולת שונה של מערכת המידע הממוחשבת לתמוך בביצועיה.

אם נדרג את הפונקציות בהתאם ליכולת מערכת המידע הממוחשבת לתמוך בהן נקבל את הדירוג האיכותי הזה:

— פונקצית הבקרה — תמיכה רבה
— פונקצית התכנון — תמיכה בינונית
— פונקצית האיוש, הארגון והניהוג — תמיכה מועטת.

קבלת החלטות (תרשים 1.4)

כאמור, אחד מהליכי הניהול הנו המרת המידע בהחלטות.
אכן, הפעילות הניהולית הבאה לידי ביטוי בכל פונקציות הניהול המאפיינת יותר מכל את תהליך הניהול כולו היא פעילות קבלת ההחלטות.
איתורן של ההחלטות הנעשות בארגון וניתוח תהליך עשייתן מהווים בסיס לאיתור צורכי המידע הנדרשים לניהול של הארגון.

את המידע בכללותו ניתן לחלק לשתי קטגוריות:

— מידע לתפעול
— מידע לניהול

המידע לתפעול מתחלק לשני סוגים:

— מידע פקידותי — מערכת המידע הממוחשבת מחליפה עבודה ידנית בעבודת מכונה. לדוגמא: הפקת תלוש השכר, הפקת הזמנה לספק וכו׳.

— מידע של פעולה — מערכת המידע הממוחשבת היא שמקבלת את ההחלטה. לדוגמא: בחירת ספק מסוים, קביעת לוח־זמנים לייצור, כספומט.

מידע לניהול — מידע התומך בתהליך קבלת ההחלטה; התהליך עצמו נתון בידיו ובאחריותו של המנהל והמידע משמש לו ככלי העשוי לשפר את טיב התהליך. ככל שתהליך קבלת ההחלטה מוגדר טוב יותר כן רבה יותר יכולתה של מערכת המידע הממוחשבת לתמוך בתהליך.

תהליך קבלת החלטה אינו פעולה בנקודת זמן. מספר שלבים מרכיבים אותו, כשלכל שלב תפקידו הייחודי והתייחסותו המוגדרת למערכת המידע הממוחשבת. לתהליך שלושה שלבים עיקריים המתפצלים למספר תת-שלבים:

— שלב הזיהוי (identification)
— שלב הניתוח והתיכון (analysis and design)
— שלב הבחירה (choice).

כל שלב עושה שימוש במידע. עקרונית, לא חייב אותו גורם לבצע את התהליך כולו בשלמותו וניתן להפקיד כל חלק מהתהליך בידי גורמים שונים (מחשב, פונקציות מטה, המנהל המחליט).

ביצוע התהליך בצורה נכונה פירושו ביצוע כל שלביו בסדר המוכתב. אולם, יש לזכור, שהתהליך הנו איטרטיבי במהותו.

כדי לעשות שימוש נכון בתהליך קבלת ההחלטה יש להכיר את שלביו:

א. שלב הזיהוי

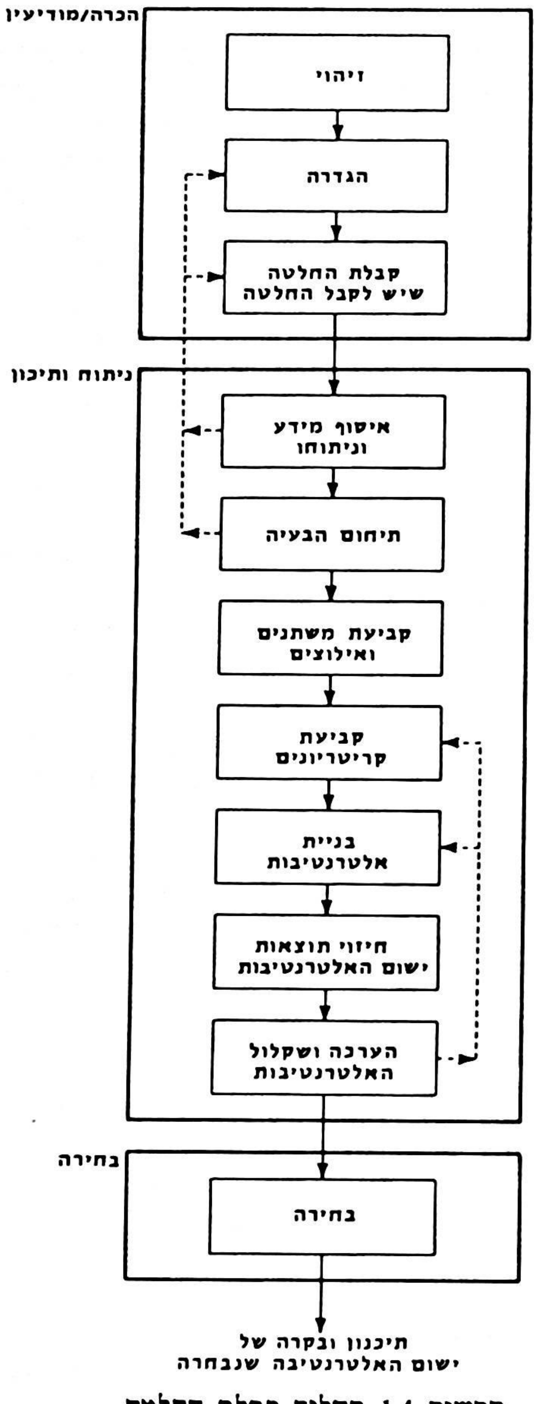
— איתור וזיהוי הבעיה או ההזדמנות (תפישה)
— הגדרת הבעיה או ההזדמנות (ניסוח)
— קבלת החלטה שיש מקום להמשיך בתהליך (קיימת בעיה ויש אפשרות לפותרה).

ב. שלב הניתוח והתיכון

— איסוף מידע רלוונטי (על הסביבה ועל המערכת שבתוכה מקבלים את ההחלטה) וניתוחו;
— הגדרת תחומי הבעיה או ההזדמנות;
— איתור המשתנים והאילוצים הרלוונטים לבעיה או להזדמנות וקביעתם;
— הגדרת הקריטריונים לפיהם ישוקללו חלופות שייבחנו;
— בניית חלופות סבירות, לאמור תיכון דרכי פעולה אפשרית שביצוען יביא לפתרון הבעיה או לניצול ההזדמנות;
— הערכת עלויות ותועלות של החלופות הנבחנות;
— חיזוי תוצאות יישום החלופות (על-ידי ניסוי, מודל וכו׳)
— השוואת התוצאות החזויות לקריטריוני הבחירה ושקלול החלופות בהתאם.

ג. בחירה

— בחירת החלופה הטובה ביותר (״הפתרון״)
— תכנון יישום הפתרון שנבחר
— בקרה על תוצאות יישום הפתרון שנבחר.



תרשים 1.4 תהליך קבלת החלטה

להלן, נקודות הבהרה על תהליך קבלת ההחלטה ועל חלקה של מערכת המידע הממוחשבת בו:

א. כל שלבי תהליך קבלת ההחלטה הכרחיים וסדר ביצועם הכרחי אף הוא (אם לא מופסק התהליך לפני סיומו). זאת, בנוסף לאפשרות שהתהליך יהיה איטרטיבי.

ב. קיימים קשרי גומלין של קלט ושל פלט בין שלבי התהליך.

ג. דרוש מידע לתהליך קבלת ההחלטה בכל שלביו.

ד. מערכת המידע הממוחשבת מסוגלת לסייע אך ורק להחלטות המוגדרות כרציונאליות. רציונאליות בהקשר זה פירושה דבקות בקריטריונים לבחירה, כלומר, יש לדבוק בהם בעת הבחירה בין החלופות. תחילה נקבעים הקריטריונים, לאחר מכן החלופות וחיזוי תוצאות יישומן, ולבסוף בחירת החלופה הטובה ביותר לאור ההשוואה בין התוצאות החזויות לבין הקריטריונים. במסגרת של קבלת החלטות רציונאליות, הבחירה החופשית קיימת רק בשלב קביעת הקריטריונים ולא בשלב הבחירה, כמובן בתנאי שניתן לסדר את החלופות בסדר התואם את הקריטריונים.

ה. חיזוי תוצאות היישום של החלופות הסבירות יכול להיעשות על־ידי ניסוי מעשי (דרך יקרה, ארוכה ומורכבת) או על־ידי בניית מודל האמור לתאר בקירוב סביר את התנהגות המערכת האמתית.

ו. בשלב ניתוח המידע הנדרש בתהליך קבלת ההחלטה, מתבצעות פעולות עיבוד הנתונים האלה:

— בדיקת עקביות הנתונים — אימות (verification)
— סידור הנתונים — מיון (sorting)
— אבחנה בין טפל לעיקר — שליפה (retrieval)
— ביצוע חישובים (computation)
— הצגת תוצאות (display).

פעולות אלו יכולות להיות מבוצעות על־ידי מערכת מידע ממוחשבת או ידנית, או על ידי שילוב של מערכת מידע ממוחשבת עם גורמים אנושיים.

סוגי החלטות

יכולתה של מערכת מידע ממוחשבת לתמוך בתהליך קבלת ההחלטה תלוי בסוג ההחלטה. את ההחלטות ניתן לסווג לשניים:

— החלטות מובנות (structured)
— החלטות שאינן מובנות (unstructured)

החלטה מובנית, היא החלטה שתהליך עשייתה הוא אלגוריתמי (algorithm), לאמור:

* לוגי
* חד־משמעי
* מוגדר לחלוטין.

החלטה זו מתקבלת בתנאי ודאות, כלומר, כל החלופות ותוצאות יישומן ידועות/כמותיות וניתן להשוותן מול מדד כמותי.
החלטה בלתי־מובנית היא החלטה שתהליך עשייתה הוא היאוריסטי (heuristic), כלומר:

* אינו מוגדר לחלוטין
* לא ניתן לאתר ולהגדיר את כל המשתנים
* לא ניתן לכמת את כל המשתנים המוגדרים.

כדי לקבל החלטות בלתי־מובנות יש לשער השערות, לעשות שימוש באינטואיציה, להעריך, להביא לשיקול דעת, להשתמש בנסיון ולזכות במזל. זו החלטה המתקבלת בתנאי אי־ודאות — אין ודאות שדרך הפעולה שעליה יוחלט היא הטובה ביותר ושאמנם תשיג את המטרה שאליה חותרים. זאת משום שלא כל החלופות ידועות ומשום שלא ניתן לחזות את תוצאות היישום של החלופות הידועות.

ההחלטות הלא־מובנות הן נפוצות יותר; ניתן להבחין בתוכן בין החלטות מובנות יותר ומובנות פחות. למעשה, שני סוגי ההחלטות שהוגדרו לעיל הן הנקודות הקיצוניות ברצף שלארכו נעות החלטות בין מובנות לחלוטין לבין בלתי־מובנות לחלוטין.

תכונת המיבניות היא תכונת היסוד של כל החלטה. החלטות מובנות הנן בנות תיכנות (מתוכנתות; programmed decisions) ואילו החלטות שאינן־מובנות אינן בנות תיכנות (unprogrammed decisions). תיכנות בהקשר זה פירושו פיתוח תהליך קבלת החלטות מוגדר, מפורט, שלם וחד־משמעי, כלומר: הכנסת תהליך קבלת ההחלטה למסגרת מוכתבת וקבועה מראש.

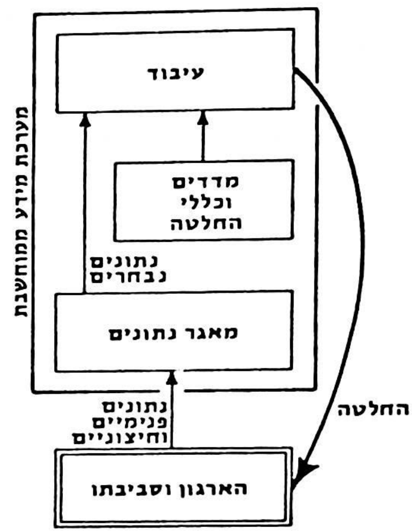
ההחלטות המתוכנתות הן בדרך־כלל ההחלטות החוזרות, הרוטיניות, ואילו ההחלטות הבלתי־מתוכנתות הן בדרך כלל החלטות חדשות, חד־פעמיות. עם זאת, לא רצוי לתכנת כל החלטה מובנית. ההחלטה אם לתכנת החלטה מובנית אם לא, חייבת להיות פועל יוצא של ניתוח עלות/תועלת.
יתירה מזו: אם החלטות אשר בתחילת קיומן הן בלתי־מובנות חוזרות על עצמן, קיימת סבירות גבוהה שעם הזמן יהפכו למובנות.

רק החלטות מובנות לחלוטין שתהליך קבלתן רציונאלי הנן בנות מיחשוב. החלטות כאלו ניתן להעביר למערכת המידע הממוחשבת (ראה תרשים 1.5א'). כל שאר סוגי ההחלטות אינן בנות־מיחשוב ובתהליך קבלתן חייב האדם להיות מעורב. מערכת המידע הממוחשבת יכולה לתרום לתהליך קבלת ההחלטה על־ידי הפקת מידע לניהול, אולם אינה מסוגלת לבצע את התהליך באופן בלעדי. תמיכה של מערכת המידע הממוחשבת יכולה לבוא לידי ביטוי בכל אחד משלושת שלבי תהליך קבלת ההחלטות.

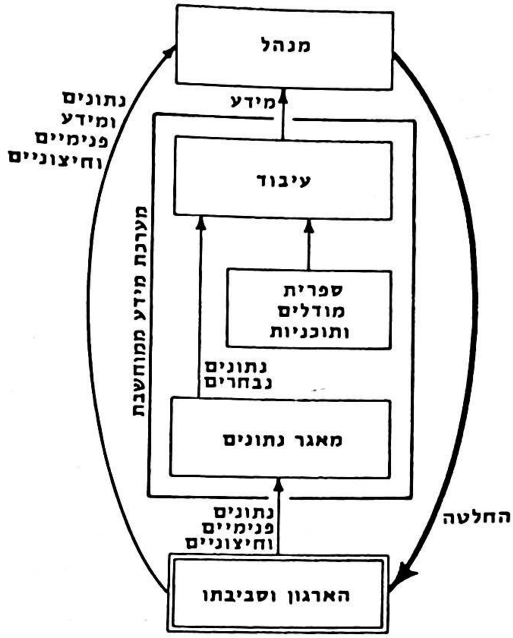
לסיכום, מערכת המידע הממוחשבת מסוגלת לקבל החלטה אם יתמלאו שני תנאים:

— תהליך קבלת ההחלטה יוגדר לחלוטין ויתוכנת
— המידע הדרוש לתהליך יימצא במערכת המידע הממוחשבת ברמות שלמות, אמינות וזמינות התואמות את צורכי התהליך.

לעומת זאת, מערכת המידע הממוחשבת יכולה רק לתמוך בתהליך קבלת החלטה שאינה מובנית. ככל שההחלטה מובנית יותר כן גדולה יכולת התמיכה של מערכת המידע. (ראה תרשים 1.5ב׳).



תרשים 1.5א׳
תהליך קבלת החלטה
מתוכנתת על־ידי מערכת
מידע ממוחשבת



תרשים 1.5ב׳
תהליך קבלת החלטה
בלתי־מתוכנתת בתמיכת
מערכת מידע ממוחשבת

מאפייני החלטות ביחס לדרג הניהול המקבל אותן

מקובל לאבחן שלושה דרגי ניהול במערכת הניהול:
— הדרג הסטרטגי (strategic planning)
— הדרג הטקטי (management control)
— הדרג התפעולי (operational control).

עקרונית, כל דרג ניהול מקבל החלטות בעלות מאפיינים יחודיים לאותו דרג. נאמר "עקרונית" משום שבמציאות אין החלוקה ברורה וחד־משמעית. לצורך ענייננו ניתן לומר שקיימים שלושה סוגי החלטות:

— החלטות סטרטגיות
— החלטות טקטיות
— החלטות תפעוליות.

לכל סוג החלטה מאפייני המידע התומכים בה והמכתיבים את המבנה של מערכת המידע האמורה לסייע בהחלטות אלו.

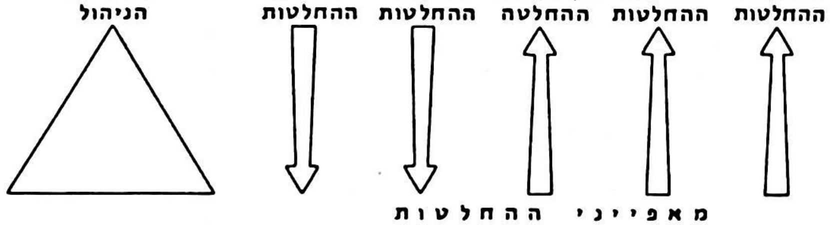
כל דרג ניהולי מבצע פעילויות ניהוליות כדלקמן:

א. הדרג הסטרטגי עוסק בהצבת מטרות הארגון ובעריכת שינויים בהן, בהחלטה על המשאבים הדרושים להשגת המטרות ובמדיניות שלפיה יעשו שימוש בהם, לאמור: הדרג הסטרטגי עיסוקו בעיקר בתכנון לטווח ארוך (למשל, הנהלה מרכזית של בנק).

ב. הדרג הטקטי עוסק בקביעת יעדים, שהם תרגום המטרות שקבע הדרג הסטרטגי, בתכנון מפורט של אופן השגתם ובבקרה על ביצוע התכנון, לאמור: תכנון לטווח קצר עד בינוני במסגרת מטרות כלליות נתונות (למשל, מנהלי מחלקות בהנהלת הבנק, או מנהלי אזורים של הבנק).

ג. הדרג התפעולי עיסוקו בניהול יומיומי של הביצוע במסגרת תהליכי העבודה, התכנון והיעדים שקבע הדרג הטקטי (למשל, מנהלי סניפים של בנק).

ניתוח פעילויות אלו מורה כי רבה יכולתה של מערכת המידע הממוחשבת לתרום לתהליכי הניהול התפעוליים, ואילו לתהליכי הניהול הסטרטגיים תרומתה האפשרית מעטה (ראה תרשים 1.6).



תרשים 1.6 אפיון איכותי של ההחלטות ביחס לדרג הניהול שבו הן נעשות (ראש החוץ מבטא "הרבה", זנבו — "מעט")

תכונות מערכת המידע בהתאם לסוג ההחלטות שהיא תומכת בהן

התכונות העיקריות המעצבות את מערכת המידע הממוחשבת הן:

— מקור הנתונים — האם המקור מצוי בתוך מערכת הארגון או מחוצה לה;
— מידת העדכניות של הנתונים האגורים במערכת. עדכניות פירושה: פער הזמן בין מועד התרחשות האירוע לבין מועד הזמינות של המידע על הארוע;
— אורך חייהם של הנתונים, לאמור: כמה זמן לאגרם;
— מידת התמצות, לאמור: באיזו רמת פירוט לאגור את הנתונים;
— מידת ההיקף, לאמור: באיזו מידה מקיפה המערכת את הארגון כולו ואת חלקיו;
— מהירות הגישה, לאמור: פער הזמן העובר בין מועד הדרישה למידע לבין מועד קבלתו;
— צורת הגישה, לאמור: טופס ידני, מודפס במכונה, שאילתא.

הטבלה הבאה מציגה את הקשר שבין תכונות המידע לסוגי ההחלטות בארגון:

סוג ההחלטה / מאפייני המידע	החלטה סטרטגית	החלטה טקטית	החלטה תפעולית
עדכניות	בינונית	גבוהה	גבוהה מאוד
אורך חיים	ארוך	בינוני	קצר
תמצות	גבוה מאוד	בינוני	אין תמצות
היקף	רחב מאוד	מוגבל	צר
מהירות גישה	אטית	בינונית	מהירה
צורת גישה	שאילתות דוחות אד־הוק דו־שיח אדם־מחשב	דוחות שוטפים דוחות חריגים שאילתות	שאילתות
מקור הנתונים	בעיקר חיצוני	בעיקר פנימי	פנימי

למידע המופק ממערכת המידע יש ערך רק כאשר עושים בו שימוש. במידע ייעשה שימוש רק אם טיבו יתאים לצרכים שלשמם הוא מופק. לפיכך, חובה לאתר את

צורכי המידע טרם בנייתה של מערכת המידע ולבחון את המידע האמור להיות מופק על ידה לפי המאבחנים האיכותיים האלה:

— רלוונטיות: האם קיים קשר בין המידע המופק לבין המטרה שלמענה הופק?
— צורה: האם צורתו של המסמך נושא המידע מאפשרת לאדם שיקבלו להבין את תוכנו?
— עיתוי: האם המידע מופק בזמן הנכון?
— יעד: האם המידע מופנה לבעל־התפקיד הנדון?
— תמצות: האם מידת התמצות תואמת את הצרכים?
— כדאיות: האם ערך המידע המופק שווה את עלותו?
— שלמות: האם המידע מקיף את הצרכים שלשמם הופק?
— אמינות: האם דיוקו של המידע המופק תואם את הצרכים?

המטרה היא שההתאמה בין הצרכים לבין המידע המופק ממערכת המידע תהיה הטובה ביותר ובה בשעה יישמר האיזון בין עלות השגת המידע והתועלת המופקת ממנו.

המבנה של מערכת מידע ממוחשבת

את מבנה מערכת המידע הממוחשבת אפשר לתאר משני היבטים שונים:

— **היבט המבנה התפקודי**
— **היבט המבנה הפיסי**

א. היבט המבנה התפקודי

מערכת המידע הממוחשבת בנויה מחמש חוליות (modules) תפקודיות לוגיות:

1. החוליה האוספת, הרושמת, והמאמתת את נתוני הקלט (data gathering module).
2. החוליה האוגרת את הנתונים (data base).
3. החוליה שבה מצויים התכניות והמודלים המופעלים על הנתונים להפקת מידע (model and program base).
4. החוליה המפיקה דוחות שוטפים ומיוחדים (report module).
5. החוליה שבאמצעותה ניתן לקבל תשובות לשאילתות ושבאמצעותה יכול המנהל להידבר עם מערכת המידע (query module).

[illegible]

[illegible]:

[illegible] — [illegible], אבצעי [illegible], [illegible].

[illegible] — [illegible], תוכנת [illegible], תוכנת [illegible], [illegible] וכו'.

[illegible] — [illegible], בה ובתי, [illegible], ניתן להבחין בין היתר בנהלי [illegible].

[illegible] — [illegible] מערכות מידע, [illegible], בפצילים [illegible].

[illegible] בארגון חון

[illegible];

[illegible] את האיסוף הרלוונטי עבורה;

[illegible] את כל הפונקציות.

[illegible] הממוחשבת ניתן לסווג בצורה דלהלן:

א. [illegible]

[illegible] ללקוחות, תלושי שכר לעובדים, הזמנות לספקים.

ב. [illegible]

[illegible] את שהתרחש בתקופה מסוימת או את המצב הקיים בנקודת [illegible] להפקת הפלט הוא הזמן (פעם ביום, פעם בשבוע וכו').

ג. [illegible]

[illegible] של חריג, כפי שהוא מוגדר על־ידי התכנית. המניע [illegible] אלה הוא האירוע ולא הזמן כפי שהדבר קיים בדוחות [illegible].

ד. [illegible] ותשובות לשאילתות

[illegible] להיות מתוכננות מראש או בלתי מתוכננות מראש. השאילתות [illegible] שאינו מצוי במסגרת הדוחות הרגילים (תקופתיים או [illegible]) או שהצורך בו אינו בר־חיזוי מראש, כמו למשל מצבו של פריט מסויים [illegible].

[illegible] לשאילתא יכולה להינתן בצורה מקוונת (on-line) מידית או לאחר [illegible] בהתאם למבנה מערכת המידע, אופי החומרה והתוכנה שלה. [illegible] באים לענות על צורך שמתכנני המערכת לא צפו מראש (הדוח

החד־פעמי יכול עם הזמן ליהפך לדוח שגרתי). במקרה זה יש להבחין בין המידע הניתן להפקה מתוך מאגר הנתונים המצוי במערכת לבין המידע שלשם הפקתו יש צורך באיסוף נתונים מיוחד (או שילוב של שני מקרים אלה).
בניית תכנית השאילתא או בניית התכנית המפיקה את הדוח החד־פעמי יכולה להיעשות על־ידי היחידה המקצועית או על־ידי המשתמש עצמו.

ה. דו־שיח אדם/מחשב

זהו מקרה שבו המשתמש יכול להגיע בעזרת דו־שיח מקוון עם ספריות המודלים, התכניות ומאגר הנתונים למידע הנדרש לפתרון בעיה. דו־שיח אדם/מחשב מכתיב שימוש במסוף ובקווי תקשורת ומצריך את קיומן של ספריות תכניות, מודלים ומאגר נתונים ברי־גישה המכילים את הדרוש למשתמש מציג הבעיה.

חלק שני

טכנולוגיה

פרק 2.

מערכת המחשב

מערכת המחשב מורכבת משני מרכיבים עיקריים, חומרה ותוכנה, הפועלים באופן משולב.

חומרה (HARDWARE)

מכלול פריטי הציוד במערכת ממוחשבת נקרא חומרה. מרכיב החומרה המרכזי הוא המחשב, השולט בכל פריטי הציוד.

מחשב (COMPUTER)

המחשב הוא התקן אלקטרוני המסוגל לפעול לפי תכנית, לבצע סדרת פעולות עיבוד אריתמטיות ולוגיות, וכן לבצע פעולות העברה ועריכה של נתונים.

המחשב, יחד עם פריטי ציוד נוספים במערכת המחשב, מסוגל לקלוט נתונים, לאחסנם ולשולפם בעת הצורך. הוא מסוגל להעביר, לעבד ולערוך נתונים, ולהציגם בצורות הניתנות להבנה ע״י אדם.

מחשב ״קורא״, ״זוכר״, ״כותב״ ומבצע פעולות חישוב ופעולות לוגיות בהתאם להוראות התכנית המפעילה אותו.

השוואה בין אדם למערכת מחשב

תכונה	אדם	מערכת מחשב
כושר חשיבה	יכולת חשיבה מותנה בכושר לימוד, אינטואיציה ואינטליגנציה	אינו חושב — פועל רק בהתאם להוראות
אגירה (זיכרון)	יכולת אגירה רבה אך לא סלקטיבית. השליפה תלויה בכושר הזיכרון	יכולת אגירה גדולה וסלקטיבית. השליפה מדויקת
מהירות	אטית	מהירות אלקטרונית גבוהה
דיוק בביצוע חישובים ופעולות	בינוני — בלתי יציב	מוחלט
כושר עבודה	מוגבל	מכונה
נקיטת עמדה	סובייקטיבי	מכונה

החומרה אינה יכולה לפעול לבד. יש צורך בסדרה של תכניות או בהוראות, אשר מהוות ביחד את התוכנה. התכנית המרכזית בחומרה נקראת מערכת ההפעלה. מערכת ההפעלה היא גם ה״קשר״ שבין האדם למכונה, ובין המכונה לתכניות היישומיות.

כדי להמחיש את האמור לעיל נשתמש בדוגמה של מטוס נוסעים. כשלעצמו, אין למטוס כל ערך. כדי לתפעל את המטוס יש צורך בטייס אשר יטיסו ע״י הפעלת כל מרכיביו. כך הדבר גם ביחס למחשב: יש צורך במערכת הפעלה כדי להפעיל את החומרה, שלעצמה אין לה כל ערך שימושי. אולם למטוס המונע ע״י טייס אין ערך שימושי, אלא אם יימצאו נוסעים במטוס אשר יועברו ליעדם. המשתמש במחשב, אם כן, גם הוא צריך לתוכנה נוספת על מנת להפיק תועלת, בדומה לנוסעי המטוס. תוכנה זאת היא התוכנה היישומית.

סוגי מחשבים

מקובל לחלק מחשבים לשתי קבוצות עיקריות: מחשב אנלוגי (analog computer) ומחשב ספרתי (digital computer): הסוג השני הוא הנפוץ יותר, וכאשר מדברים על מחשבים מתכוונים בדרך־כלל למחשבים ספרתיים.

מחשב ספרתי

מחשב ספרתי מעבד נתונים המיוצגים על ידי הימצאות או אי־הימצאות של מטען חשמלי. מחשב זה מבצע חישובים באמצעות מספרים המבטאים ערכי משתנים וכמויות. החישובים המספריים מאפשרים להגדיר כמויות באופן ספרתי בכל מידת דיוק מבוקשת. המחשבים הנמצאים בשימוש למטרת עיבוד נתונים הם בדרך־כלל מחשבים ספרתיים.

מחשב אנלוגי

מחשב אנלוגי מקבל כקלט, מעבד ומפיק כפלט נתונים רציפים בטבעם. שימושיו של המחשב האנלוגי מיועדים בעיקר ליישומים בעלי אופי פעולה רציף, כגון תהליכי בקרה, מדידות חום ולחץ. לדוגמה: כאשר אחות מודדת את חום החולה היא משתמשת במכשיר אנלוגי; מידת החום עשויה להשתנות באופן רציף. כאשר האחות מודדת את קצב פעימות הלב, היא מבצעת תהליך ספרתי בדיד, כיון שהיא מבצעת ספירה של פעימות בודדות. דוגמה אחרת היא מד־הדלק במכונית — המחוג הוא מכשיר אנלוגי המראה באופן רציף את גובה פני הדלק במכל; מנורת האזהרה היא מכשיר דיגיטלי המראה שגובה פני הדלק נמצא מעל לגובה מסוים (המנורה כבויה) או מתחת לגובה הנ"ל (המנורה דולקת).

נתונים, עיבוד נתונים ומידע

נתונים (data)— הם עובדות גולמיות בלתי מעובדות.

מידע (information) — פירושו נתונים מעובדים שאורגנו בצורה משמעותית לגבי צורך מסוים.

עיבוד נתונים אלקטרוני — ענ"א (EDP — Electronic Data Processing) — הוא תהליך בו המחשב מקבל נתונים כקלט, מארגנם, מעבדם ומוציאם כמידע — הפלט של התהליך (ראה תרשים 2.1).



תרשים 2.1: תהליך עיבוד נתונים אלקטרוני

דוגמאות מוחשיות יותר לתהליך קלט — עיבוד — פלט מוצגות להלן:

	קלט	עיבוד	פלט
אפיית עוגה	מרכיבים שונים	ערבוב ואפייה	עוגה
תהליך כביסה	בגדים מלוכלכים + אבקת כביסה	כביסה במכונה	בגדים נקיים

כדוגמה להפקת מידע מנתונים נתבונן בתהליך מציאת הכנסות של חברה מתוך נתוני מכירה שוטפים (תרשים 2.2).



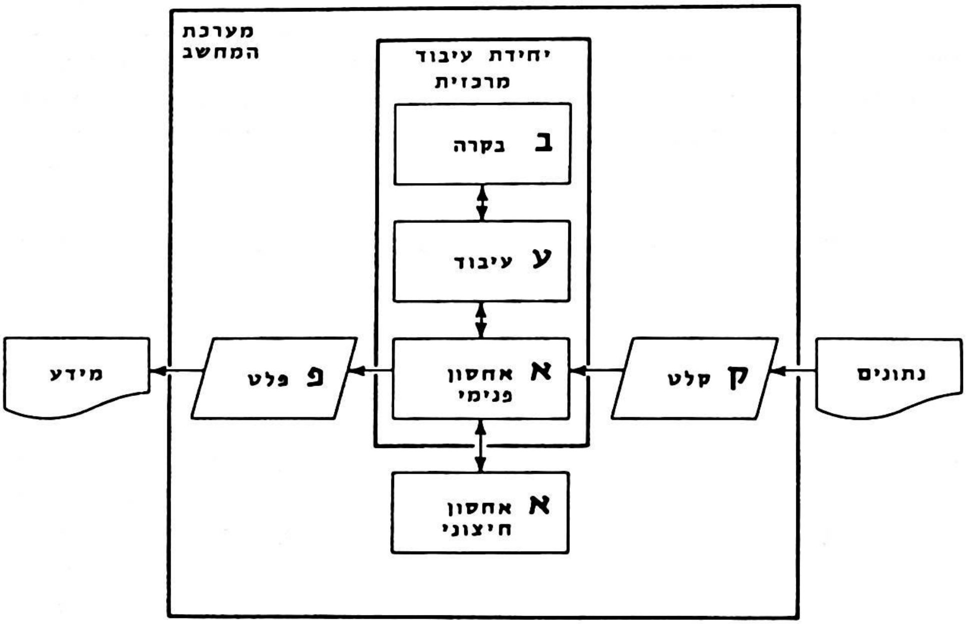
תרשים 2.2: דוגמה

תהליך "קאעבפ"

המרכיבים היסודיים של תהליך עיבוד הנתונים האלקטרוני הם (ראה תרשים 2.3):

1. **קלט** (input) — הנתונים נקראים באמצעות התקנים שונים במערכת המחשב, כגון קוראת כרטיסים, לוח מקשים.
2. **אחסון** (storage) — הנתונים שנקראו מאוחסנים במערכת המחשב בזיכרון הפנימי או ביחידת אחסון חיצונית.
3. **עיבוד** (processing) — עיבוד הנתונים מתבצע ביחידה האריתמטית-לוגית של המחשב.
4. **בקרה** (control) — הבקרה מתבצעת באמצעות יחידת הבקרה והפיקוח של המחשב.
5. **פלט** (output) — מידע המופק באמצעות התקנים שונים במערכת המחשב כגון מדפסות, מסכים, תקליטים, סרטים וכו'.

פונקציות אלו קיימות בכל תהליך עיבוד נתונים. בין במחשב, בין במערכת ידנית ובין במערכת הפשוטה ביותר — עיפרון ונייר.



תרשים 2.3: תהליך קאעב״פ

פעולת המחשב

נתוני קלט ופלט במערכת מחשב מוצגים בשיטת המספרים העשרוניים (decimal code) ובתווים אלפבתיים. לעומת זאת הפעולות הפנימיות במחשב נעשות בצופן שונה, בדרך־כלל בצופן הבינארי (binary code) (ראה תרשים 2.4).



תרשים 2.4: המרה מצופן לצופן במחשב

כדי להבין את פעולתו הפנימית של המחשב, יוגדרו תחילה מספר מונחים בסיסיים:

הסיבית (תג, BIT)

הסיבית (ראשי תיבות של "ספרה בינרית") היא המרכיב היסודי של כל מערכת מחשב ספרתי. היא מייצגת "נתח" בודד של מידע. לדוגמה: נוכחותו או היעדרו של חפץ במקום מסוים. המידע מיוצג על ידי הימצאות סיבית באחד מתוך שני מצביה האפשריים בלבד. מחשב מעבד נתונים על ידי טיפול בקבוצות של סיביות. במחשב יש מעגלים אלקטרוניים הידועים כמעגלי סב־סוב (flip־flop), שלהם שני מצבים אפשריים. הסיבית מיוצגת באמצעות מצבי הסב־סוב, לדוגמה: נורה — אם היא "דולקת" היא מציינת "לילה";

— אם היא "כבויה" היא מציינת "יום". כך מוגדר חלק היממה.

שים לב — מתוארים רק שני מצבים! שעת היום, למשל, אינה מצויינת.

תו (character)

התו מורכב ממספר סיביות. לתו צורות אפשרויות רבות, שלא כסיבית המוגבלת לשתי צורות בלבד. תו יכול להיות ספרה, אות מאותיות האל"ף־בי"ת, או סימן מיוחד, כמו רווח (space), סימן ניקוד או סימן אלגבראי.

תו מיוצג במחשב לרוב בשמונה סיביות. לעתים קרובות משתמשים בתו ובספרה כמלים נרדפות.

לדוגמה: המספר 7 הוא תו (ספרה), ואילו המספר 17 הוא מספר המורכב משני תווים.

מלה (word) של מחשב היא צירוף של תווים. המלה 3456 מכילה ארבעה תווים ונקראת נומרית. המלה ax56L8 מכילה שישה תווים. מלה כזאת נקראת אלפא־נומרית, כיון שיש בה מספרים וגם אותיות אלפביתיות. המלה acdb מכילה ארבעה תווים ונקראת אלפביתית, כיון שיש בה רק אותיות אלפביתיות.

הצופן הבינרי

הגדלת מספר המצבים הניתנים להצגה בצופן הבינרי נעשית על ידי הוספת סיביות. למשל, שימוש ב־2 נורות מאפשר ציון של 4 מצבים.

	נורה 1	נורה 2
מצב 1:	כבויה	דולקת
מצב 2:	דולקת	כבויה
מצב 3:	כבויה	כבויה
מצב 4:	דולקת	דולקת

כך למשל אפשר לתאר, בוקר, אחר־הצהריים, ערב, לילה. משתמשים ב־1 וב־0 כ"שמות" נוחים למצבים מנוגדים הניתנים להבחנה (on ו־off); האפס מציין, כמובן, את המצב הפחות פעיל של המכשיר (הנורה — במצב כבוי). היחס בין מספר המכשירים הדו־מצביים, או הסיביות, לבין מספר התנאים

השונים הוא יחס מעריכי (exponential). פירוש הדבר ש־2 בחזקת מספר הסיביות שבשימוש קובע את מספר האפשרויות, באופן הזה:

עם 2	סיביות	מספר	המצבים הוא	$4 = 2^2$
" 3	"	"	"	$8 = 2^3$
" 4	"	"	"	$16 = 2^4$
" 5	"	"	"	$32 = 2^5$
" 10	"	"	"	$1024 = 2^{10}$
" 20	"	"	"	$1,048,576 = 2^{20}$

יחס זה הוא בסיסי לצופן הבינרי.

אם נשתמש ב־10 האצבעות של כפות ידינו, הרי שבשיטה העשרונית נוכל לספור עד 10 בלבד. אבל אם בשיטה הבינרית ניתן לאצבע הראשונה את הערך "1" ולאצבע השניה ערך כפול — "2", ולשלישית ערך כפול מהשנייה — "4", וכך הלאה, נראה שבעשר אצבעות נוכל לספור עד 1024 (2^{10}).
שים לב — סך הכול האצבעות הוא 1023, אבל "0" נחשב גם הוא.

אצבע:	1	2	3	4	5	6	7	8	9	10
ערך:	1	2	4	8	16	32	64	128	256	512

למשל:

את המספר 3 אפשר לסמן על־ידי 2 אצבעות בלבד: 1 ו־2
את המספר 4 אפשר לסמן על־ידי 1 אצבע בלבד: 3
את המספר 5 אפשר לסמן על־ידי 2 אצבעות בלבד: 1 ו־3
את המספר 17 אפשר לסמן על־ידי 2 אצבעות בלבד: 1 ו־5
את המספר 856 אפשר לסמן על־ידי 5 אצבעות בלבד: 4, 5, 7, 9, ו־10

צופן אוקטלי (octal)
שיטה זו מבוססת על בסיס 8. ייצוג הספרות 7—0 נעשה כבשיטה העשרונית. שיטה זו מכילה רק 8 ספרות.

7, 6, 5, 4, 3, 2, 1, 0 ← בסיס 8

הספרה הגדולה ביותר בשיטה זו היא 7. הספרה העשרונית 8 תיוצג בשיטה זו על־ידי המספר האוקטלי 10.

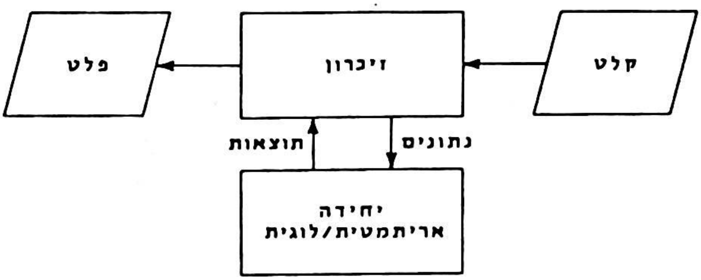
צופן הקסאדצימלי (hexadecimal)
שיטה זו מבוססת על בסיס 16. ייצוג הספרות 9—0 נעשה כבשיטה העשרונית, אולם כדי לייצג את הערכים העשרוניים 15—10 משתמשים באותיות A—F.

לדוגמה: הערך העשרוני 10 מיוצג בשיטה זאת על־ידי האות A, 11 מיוצג באות B וכך הלאה. שיטה זו משתמשת ב־16 סימונים.

F, E, D, C, B, A, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0 ← בסיס 16

דרכי החישוב של המחשב

המחשב מעבד נתונים על־ידי אחסנת הנתונים ושליפת חלקים מהם כאשר הם נדרשים לחישוב. החישוב עצמו נעשה ביחידה האריתמטית/לוגית (תרשים 2.5). יחידה זו מסוגלת לבצע פעולות חיבור בלבד. כל שאר הפעולות האריתמטיות מבוצעות באמצעות פעולת החיבור.



תרשים 2.5: דרך החישוב של המחשב

פעולת החיבור במחשב

דוגמה: פעולת החיבור של המספרים 163 ו־265 תתבצע בשלושה צעדים:

	1	6	3	
+				
	2	6	5	
			8	צעד א'
		2(1)		
+	1			צעד ב'
	4			צעד ג'
	4	2	8	

חיסור על־ידי חיבור

קל לחסר מספר ממספר שכל אחת מספרותיו היא 9. לדוגמה, קל להפחית 163 מ־999. התוצאה, 836 היא המשלים ל־9 של המחסר 163. לצורך חיסור על־ידי חיבור נבצע 4 שלבים:

1. חשב את המשלים ל־9 של המחסר.
2. חבר למחוסר את המספר המשלים שהתקבל.
3. הוסף לתוצאה 1.
4. בסכום שהתקבל הזנח את הספרה השמאלית (החשובה), אשר תהיה תמיד 1.

התוצאה תהיה ההפרש המבוקש.

דוגמה:

```
  265
— 163
─────
  102
```

כדי לחסר 163 מ־265, נחשב את המספר המשלים של 163: (חישוב המשלים הוא פעולה אלקטרונית המבוצעת על־ידי המעבד).

```
  999
— 163
─────
  836
```

נוסיף למחוסר את התוצאה:

```
  265
+ 836
─────
 1101
```

חיבור 1 והזנחת הספרה השמאלית נותנים לנו את המספר 102, שהוא התשובה הנכונה.

כפל באמצעות חיבור

דוגמה:

```
  265
x 163
─────
  795
 1590
 265
─────
43195
```

למעשה, אנחנו אומרים: 3 פעמים 265 = 795, מזיזים טור אחד שמאלה, 6 פעמים 265 = 1590, מזיזים טור נוסף שמאלה, ופעם אחת 265.

לכן אפשר לבצע זאת בצורה הזו:

```
     265
   x 163
   -----
   + 265
   + 265
   + 265
  + 265
  + 265
  + 265
  + 265
  + 265
  + 265
    265
  -----
   43195
```

ההכפלה במחשב נעשית כך: המספר המוכפל פשוט מהובר לעצמו כמה פעמים, כמספר המופיע בספרת היחידות של הכופל. המחשב זז לאחר מכן טור אחד שמאלה, וכן הלאה.

חילוק באמצעות חיבור

גם חילוק אפשר לבצע באמצעות חיבור.

דוגמה: נניח כי יש לחלק את המספר 33 ב־9. נחסר 8 מ־33 ונבחן אם תוצאת הביניים היא חיובית או אפס. אם התוצאה חיובית, אנו מוסיפים 1 למונה התוצאה (מונה זה הוא מקום מסוים במחשב, שנקבע לצבירת התוצאות). תהליך זה חוזר ושונה עד אשר מתקבלת תוצאה היובית הקטנה מ־8. אם תוצאת הביניים היא 0, מסתיים תהליך החילוק והתקבלה תוצאה מדויקת.

33	מונה
— 8	
25 חיובי	0 + 1 = 1
— 8	
17 חיובי	1 + 1 = 2
— 8	
9 חיובי	2 + 1 = 3
— 8	
1 חיובי קטן מ־8	3 + 1 = 4

התוצאה הסופית של חילוק נמצאת אפוא בין 4 ל־5.
דיוק החילוק: כדי לקבל דיוק רב יותר, קובעים את הנקודה העשרונית מימין לתוצאה שהתקבלה, כופלים את תוצאת הביניים ב־10 וממשיכים עד לדיוק הרצוי.
נמשיך את הדוגמה הקודמת כדי לקבל תוצאה מדויקת יותר:

10 = 10 x 1		מותה
8 −		
2	חיובי קטן מ־8	1 = 1 + 0

כופלים ב־10 וממשיכים:

20 = 10 x 2		מונה
8 −		
12	חיובי	1 = 1 + 0
8 −		
4	חיובי קטן מ־8	2 = 1 + 1

כופלים ב־10 וממשיכים:

40 = 10 x 4		מונה
8 −		
32	חיובי	1 = 1 + 0
8 −		
24	חיובי	2 = 1 + 1
8 −		
16	חיובי	3 = 1 + 2
8 −		
8	חיובי	4 = 1 + 3
8 −		
0	אפס	5 = 1 + 4

התוצאה — 8.125

אם אין מורים למחשב להפסיק את הפעולה כאשר נתקבלה התוצאה הנכונה (נתגלתה שארית של אפס), ימשיך המחשב בפעולה עד אשר יושג גבול הקיבולת של מונה התוצאה.
לדוגמה: אם מונה הוא בן 11 ספרות, שאחת מהן מיועדת לסימון ערכו השלילי או החיובי של המספר, תהיה התשובה לחילוק 33 ב־8 4.125000000+. ברור ששיטה זו של חילוק באמצעות חיסור (למעשה באמצעות חיבור) גוזלת זמן רב. למשל, לביצוע "1000 לחלק ב־2" צריך לעשות 500 פעולות.

חשבון בינרי

כללי החיבור של חשבון בינרי:

1. 1 + 1 = 0 ונגררת של 1
2. 1 + 0 = 1
3. 0 + 0 = 0

חשוב לציין שהשיטה הבינרית דורשת רק 3 כללים בעוד שהשיטה העשרונית דורשת הרבה יותר. לפני שנביא דוגמה, חשוב להציג את השיטה של ההמרה בכל מספר בינרי:

הטור הימני הוא טור היחידה 1 ($2^0 = 1$)
הטור השני מימין מייצג 2 ($2^1 = 2$)
הטור השלישי מימין מייצג 4 ($2^2 = 4$)
הטור הרביעי מימין מייצג 8 ($2^3 = 8$)
וכן הלאה.

באופן כזה המספר העשרוני 7 מורכב מ־ 4 + 2 + 1 = 7
והמספר העשרוני 9 מורכב מ־ 1 + 8 = 9

נניח כי מבקשים לחבר 6 ו־3 בצופן הבינרי:

בינרי		עשרוני
0110	=	6
+ 0011		+ 3
1001		9

נבדוק את התוצאה:

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

$$8 \quad 0 \quad 0 \quad 1 \quad = 9$$

חיבור סדרתי וחיבור במקביל

חיבור סדרתי הוא הדרך המקובלת לחיבור מספרים: מחברים ספרה ספרה. כאשר המהירות חשובה משתמשים בחיבור במקביל, אשר לפיו מחברים את כל הספרות בו־זמנית. ההבדל הוא עקרוני ביותר, וראוי להבחין בכך.

חיבור סדרתי

	4	3	6
	+2	1	7
צעד א'			3(1)
		+1	
צעד ב'		5	
צעד ג'	6		
	6	5	3

חיבור במקביל

	4	3	6
	+2	1	7
צעד א'	6	4	3(1)
צעד ב'		+1	
	6	5	3

בשיטת החיבור במקביל במספרים גדולים החיסכון במספר הפעולות הוא רב.

מערכת המחשב

אם תעשיית המטוסים היתה מתפתחת בקצב של תעשיית המחשבים ב־25 השנים האחרונות, היה מטוס בואינג 767 עולה כיום כ־500 דולר, היה מקיף את העולם בכ־20 דקות והיה משתמש בכעשרים ליטר דלק. דוגמה זו ממחישה יותר מכול את הגידול במהירות עבודת המחשבים, את הירידה במחירם ואת הירידה בצריכת האנרגיה שלהם. מחיר מרכיבי המחשב יורד בקצב של 25 עד 40 אחוז בשנה; מהירות החישוב גדלה פי 200 בעשרים וחמש השנים האחרונות; במשך אותה תקופה קטנה צריכת האנרגיה של מחשבים בעלי אותה עצמה ביחס של 10,000.

קצב ההתפתחות והחידושים בתחום הטכנולוגיה מהיר מאוד, ועל כן חלק מהנושאים האמורים כאן עשוי להתיישן כבר בעת כתיבתו.

מערכת מחשב מורכבת מפריטי ציוד רבים ומגוונים, הנבחרים בהתאם לצורכי עיבוד הנתונים של הארגון שבו מותקנת המערכת. (תרשים 2.6). תהליך עיבוד הנתונים מחייב קיומן של מספר פונקציות: קלט, אחסון, עיבוד, בקרה ופלט (ראה תרשים 2.3). כל פונקציה כזו מתבצעת באמצעות פריט ציוד מסוים. הבקרה, הפיקוח וחלק מצומצם של האחסון מרוכזים ביחידת העיבוד המרכזית,

יחידת הבקרה מפקחת על יחידות הקלט וקובעת אלו נתונים לקרוא, מהיכן, ובאיזה מקום בזיכרון הפנימי לאחסן את הנתונים הללו. יחידה זו מורה ליחידה האריתמטית־לוגית אילו פעולות לבצע על הנתונים, מאיזה מקום בזיכרון לשלוף אותם, והיכן לאחסן את התוצאות. לבסוף, יחידת הבקרה גם מפקחת ופוקדת על אמצעי הפלט מה להפיק והיכן. כל זה נקבע בהתאם להוראות התכנית המבוצעת באותה עת.

היחידה האריתמטית־לוגית מבצעת את פעולות החישוב וההשוואה (העיבוד) של הנתונים בהתאם להנחיות התכנית.

הזיכרון הפנימי מאחסן את הוראות הפעולה של התכנית ואת הנתונים המיועדים לעיבוד על־ידי התכנית. זיכרון זה מאופיין בזמן גישה קצר לנתונים ובמהירות העברת נתונים גבוהה מאוד. הזיכרון הפנימי בנוי מתאי־זיכרון, המורכבים ממספר סיביות. מספר הסיביות בתא־זיכרון קובע את גודל מלת המחשב (כזכור, מלת המחשב היא צירוף של תווים). קיבולת הזיכרון נקבעת על־פי מספר התאים המוכלים בו. לכל תא כתובת המאפשרת לזהותו בעת הצורך. התכנית יכולה לקרוא או לכתוב נתונים בתאי־הזיכרון האלה על־ידי הוראות מתאימות. קיבולו של הזיכרון מצומצם יחסית ואיננו מאפשר אחסון של כל הוראות הפעולה והנתונים הדרושים לביצוע תכנית, לכן מאוחסנים רק חלק מהוראות הפעולה של התכנית וחלק מהנתונים — אלה הדרושים לעיבוד מיידי. הוראות פעולה ונתונים שאינם דרושים לעיבוד מיידי נשמרים ביחידות האחסון החיצוני. כאשר הם נדרשים לעיבוד, הם מועברים אל הזיכרון הפנימי, ובגמר העיבוד הם מועברים חזרה אל יחידות האחסון החיצוני.

הזיכרון הפנימי בנוי מתאים המייצגים את הסיביות והניתנים למיגנוט: המיגנוט יכול להיות באחד משני מצבים, 0 או 1. מהירות העברת הנתונים בזיכרון הפנימי הנה כמהירות זרימת החשמל (מהירות האור); זמן הגישה לתא זיכרון מסוים תלוי בגודלו הפיסי של הזיכרון, כלומר — במרחק שזרם החשמל צריך לעבור בין תאי הזיכרון כדי להגיע לתא הרצוי.

טכניקות שונות וחומרים שונים משמשים לבניית הזיכרון הפנימי במטרה לצמצם את גודלו.

קיימים שני סוגים של זכרונות:

א. Read Only Memory (ROM) — זהו זיכרון המכיל הוראות אשר תוכנתו על־ידי יצרן המחשב ואין כל אפשרות לשנותם. זיכרון זה מוגן, כך שגם הפסקת חשמל או כיבוי המערכת אינם גורמים לאיבוד התוכנה הנמצאת בזיכרון זה.

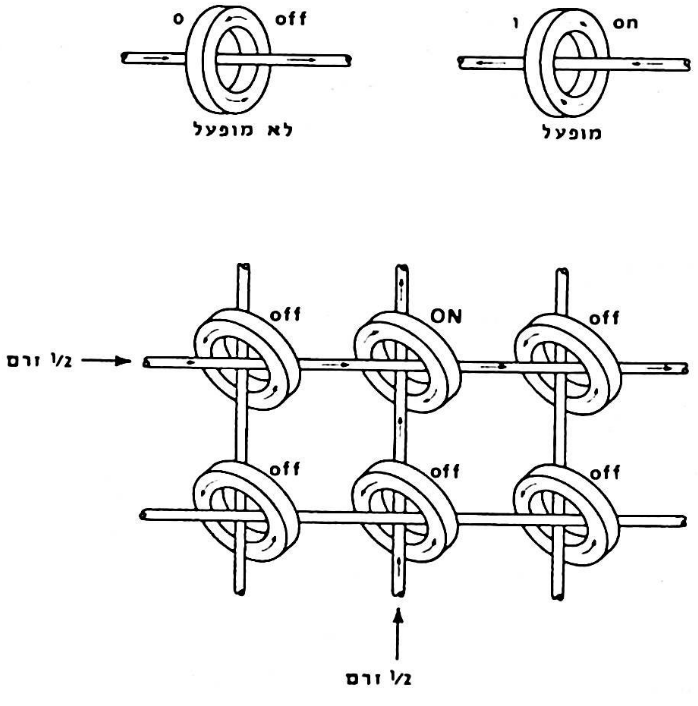
ב. Random Access Memory (RAM) — זהו זיכרון שלצורך ביצוע עבודה מסוימת הוזן בתוכנה ובנתונים על־ידי מפעיל המערכת. במקרה של הפסקת חשמל נמחק כל המידע הנמצא בזיכרון זה.

מהמוסבר לעיל ברור כי ה־ROM אינו עומד לרשות המשתמש בעוד שה־RAM הנו הזיכרון העומד לרשותו. לכן, בקביעת קיבולת הזיכרון חשוב לבחון את גודלו של הזיכרון הזה.

הזיכרון הפנימי מורכב ממעגלים משולבים (integrated circuits) הבנויים ממיליוני חלקיקי סיליקון. תפושתו קטנה במיוחד — סנטימטר מרובע מכיל עשרות אלפי סיביות. טכניקות חדשות מפותחות הומצאו, במטרה להגדיל את הקיבולת שלו, להוזיל את מחירו ולשפר את מהירות ההעברה שלו.

כדי להדגים את אופן פעולת הזיכרון הפנימי, נשתמש בזיכרון טבעות, אשר היה בעבר נפוץ בשימוש. כיום כמעט שלא נעשה שימוש בזיכרון כזה עקב גודלו, מחירו, צריכת החשמל והאטיות (היחסית).

זיכרון טבעות (magnetic core memory) בנוי ממיליוני טבעות ברזל הניתנות למיגנוט על־ידי זרם העובר דרכן. כל טבעת מייצגת סיבית אחת, שניתן לזהותה לפי מיקומה ברשת החוטים האנכיים והאופקיים. העברת זרם בעצמה מסוימת דרך שני החוטים המתאימים תמגנט את הטבעת הנמצאת בהצטלבות החוטים הללו. כיוון הזרם קובע את כיוון המיגנוט. טבעת ממוגנטת יכולה להישאר במצבה ללא הגבלת זמן (ראה תרשים 2.7).



תרשים 2.7: זיכרון טבעות

יחידות קלט ופלט (Input and Output Devices)

מסוף (terminal)

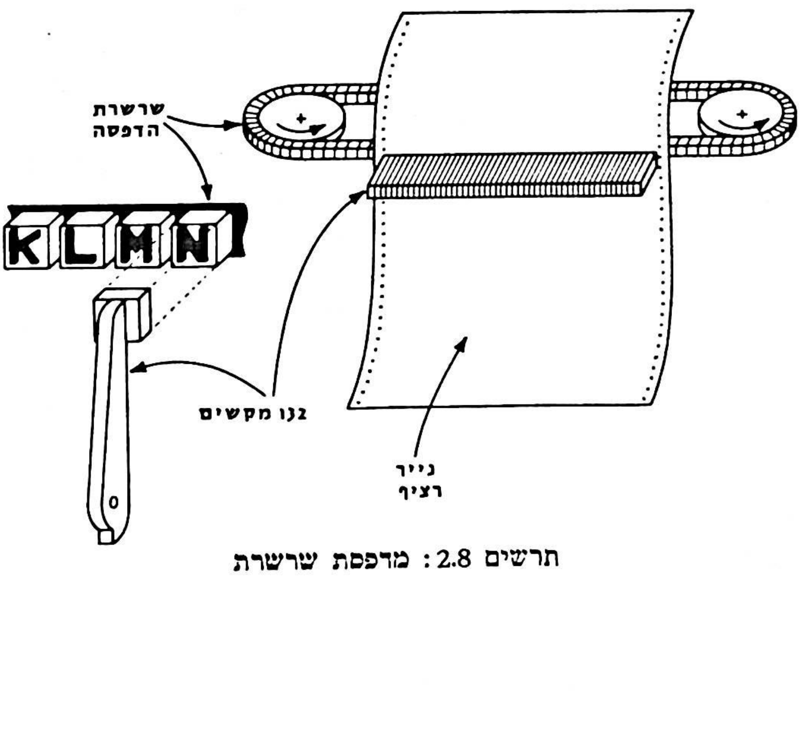
המסוף הוא אמצעי הקלט/פלט הנפוץ ביותר. המסוף מצויד בלוח מקשים, הדומה ללוח המקשים במכונת כתיבה, ובמסך או נייר שעליו מוצגות תשובות המחשב. מפעיל המסוף מקיש את הנתונים או את הוראות הפעולה על לוח המקשים, המסוף מתרגם את התווים שהוקשו לאותות חשמליים המועברים למחשב, וכאשר מתקבלת תגובת המחשב — מתרגם המסוף את האותות החשמליים לתווים המוצגים על גבי המסך או המודפסים על גבי הנייר.

מדפסות

המדפסת הופכת את האותות החשמליים המייצגים את תוצאות העיבוד לאותיות ולספרות ומדפיסה אותם על גבי נייר. קיים מספר רב של מדפסות. המדפסות נבדלות הן במהירות ההדפסה והן באיכות (resolution) ההדפסה.

מדפסת שורה (line printer)

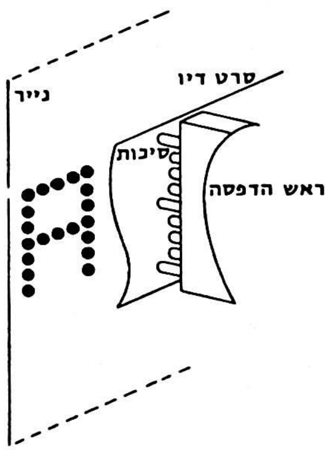
מדפסות מסוג זה יכולות להדפיס בין 300 ל־3000 שורות בדקה. מדפסות אלו פועלות על עיקרון דומה למכונת כתיבה: נייר רציף מורץ במהירות בין שרשרת ההדפסה לסרט ספוג דיו. שרשרת ההדפסה מכילה מספר סדרות של תווים ומסתובבת במהירות. המקשים מסודרים לרוחב נייר ההדפסה, בהגיע תו למקום המיועד להדפסתו על גבי הנייר מופעל מגנט חשמלי המכה על המקש ומצמיד את הנייר לסרט הדיו מול התו אותו יש להדפיס (תרשים 2.8).



תרשים 2.8 : מדפסת שרשרת

מדפסת נקודות (dot-matrix printer)

מדפסות אלו זולות יחסית ומסוגלות להדפיס עד 300 תווים לשנייה. בגלל מבנה מטריצת הנקודות ניתן להדפיס בעזרתן סוגי אותיות שונים, וכמובן, בשפות שונות. ניתן בעזרתם גם לקבל פלט גרפי. תרשים 2.9 מראה בצורה סכמטית את צורת פעולתה של מדפסת נקודות. ראש ההדפסה בנוי מקבוצה של סיכות אשר נלחצות בצורה סלקטיבית (בהתאם לאות הרצויה). כאשר הראש מתקדם בכיוון ההדפסה, הסיכות הנלחצות לוחצות על סרט הדיו (ribbon) וכך נוצרת ההדפסה. בתרשים 2.9 נראית מדפסת אשר ראש ההדפסה שלה מכיל 9 סיכות, ולכן גובה האות יהיה בנוי מתשע נקודות. רוחב האות נקבע על־ידי התכנית המפעילה את המדפסת.

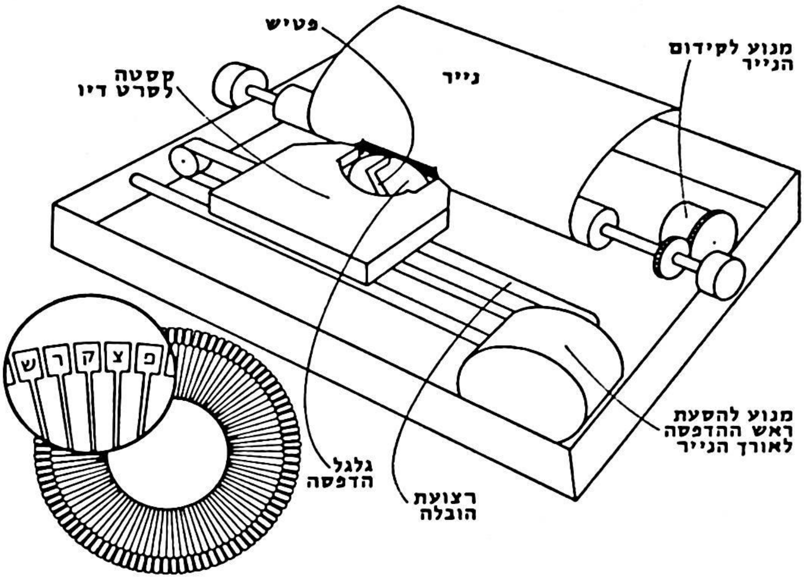


תרשים 2.9 : מדפסת נקודות

מדפסת איכות (quality printer)

מדפסות אלו דומות בעיקרון הפעולה למכונות כתיבה רגילות. ההדפסה היא באיכות גבוהה, בקצב אטי של 55—20 תווים לשנייה. גלגל ההדפסה מורכב מ־96 זרועות (במודלים מסוימים 130). בכל קצה זרוע נמצאת צורת התו (אות, ספרה או סימן אחר). תרשים 2.10 מציג ציור סכמטי של מדפסת איכות. בתחתית התרשים מוצג גלגל ההדפסה. ההדפסה נעשית על־ידי סיבוב הגלגל עד הגעתו

אל התו הרצוי והפעלת פטיש עליו הגורם להדפסתו באמצעות סרט דיו. מדפסת מסוג זה שכיחה במערכות מעבדי תמלילים.



תרשים 2.10: מדפסת איכות

מדפסות נוספות

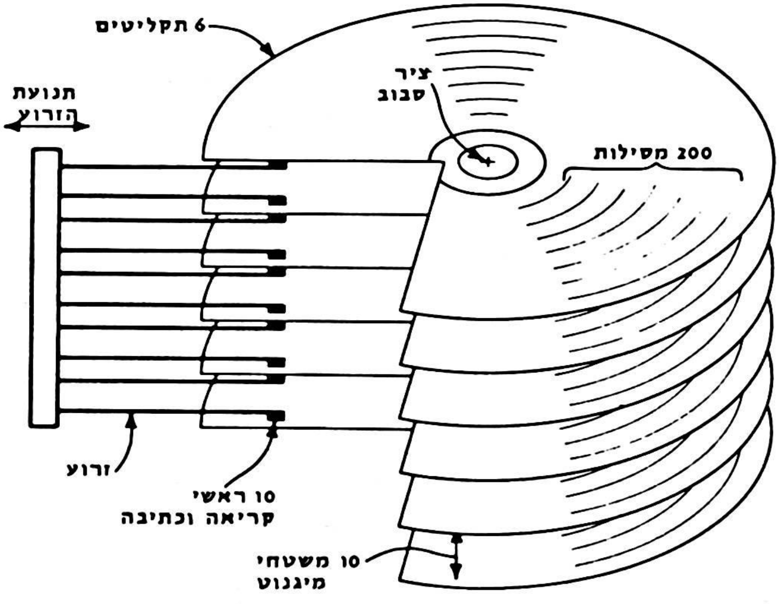
כיום קיימות מדפסות מתקדמות מבחינה טכנולוגית, המבצעות את ההדפסה באמצעות קרן לייזר (laser) או ע"י התזת דיו (inkjet). מדפסות אלו מהירות מאוד ומאפשרות הדפסה של 450 שורות לשנייה. איכות ההדפסה טובה יותר בהשוואה למדפסת השורה.

קיימות גם מדפסות המאפשרות הדפסה בצבעים שונים.

תקליט מגנטי (magnetic disk)

התקליט המגנטי בנוי ממספר תקליטים המחוברים לציר אחד. התקליטים עשויים מחומר מתכתי הניתן למיגנוט. לכל תקליט שני משטחי מיגנוט וכל משטח מחולק למסילות טבעתיות הערוכות סביב מרכז התקליט. על מסילות אלו נכתבים הנתונים. לכל משטח מיגנוט מתאימה זרוע אשר בקצה מורכב ראש הקריאה־כתיבה; הזרוע מסוגלת לנוע הן אל מרכז התקליט והן להיקפו, ולהביא את ראש הקריאה־כתיבה מול המסילה הרצויה (תרשים 2.11). קבוצת מסילות הנמצאת האחת מעל השנייה במשטחי הקריאה נקראת גליל.

התקליט המגנטי מורכב בכונן התקליטים ומסתובב במהירות גבוהה מאוד. הנתונים כתובים לאורך המסילות ונקראים או נכתבים על־ידי ראש הקריאה־כתיבה בזמן סיבוב התקליט. הנתונים נשמרים על גבי התקליט בצופן בינרי בצורת נקודות ממוגנטות ולא־ממוגנטות. יתרונותיו של התקליט המגנטי — מהירות גבוהה בקריאת נתונים ובכתיבתם, זמן גישה קצר לנתונים, קיבולת גדולה, ורישום נתונים בסדר בלתי עקיב (בניגוד לסרט המגנטי). חסרונו של התקליט המגנטי במחירו הגבוה.

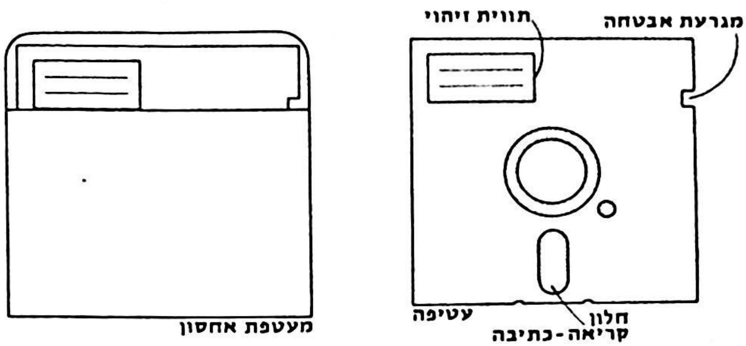


תרשים 2.11: תקליט מגנטי

תקליטון (diskette)

התקליטון הוא למעשה גירסה מוקטנת של התקליט המגנטי. מחירו זול יחסית והוא משמש כאמצעי שכיח לאחסון נתונים במיקרו מחשבים. התקליטון הממוגנט נמצא באופן קבוע בתוך עטיפת מגן. (ראה תרשים 2.12). הקריאה והכתיבה באמצעות הזרוע נעשים דרך חלון הקריאה/כתיבה. מגרעת אבטחה מגנה מפני כתיבה לא רצויה הגורמת למחיקת נתונים; כאשר המגרעת מכוסה

(בעזרת מדבקה) לא ניתן לכתוב, ואפשר לקרוא בלבד. כאשר התקליטון אינו בשימוש הוא מאוחסן במעטפת אחסון.

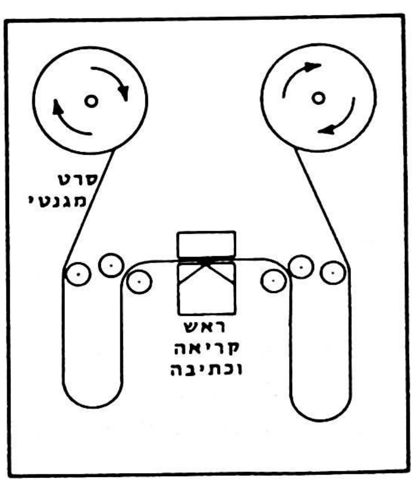


תרשים 2.12: תקליטון

סרט מגנטי (magnetic tape)

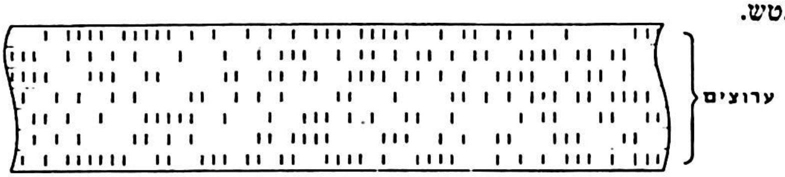
הסרט המגנטי עשוי מחומר פלסטי שצופה בחומר הניתן למיגנוט. לצורך קריאה או כתיבה על הסרט המגנטי מורכב גליל הסרט על כונן, המעביר את הסרט מול ראש קריאה־כתיבה. הכתיבה על הסרט נעשית על־ידי מיגנוט נקודות בהתאם לתווים שאותם יש לכתוב, וקריאת הסרט נעשית על־ידי קריאת נקודות אלו והפיכתן לאותות חשמליים (תרשים 2.13).

כונני הסרט המגנטי מאופיינים במהירות הקריאה והכתיבה על גבי הסרט ובצפיפות הנתונים הנכתבים על הסרט. מהירות הקריאה והכתיבה מהסרט המגנטי הנה כ־350,000 תווים בשנייה.



תרשים 2.13: כונן סרט מגנטי

הכתיבה על גבי הסרט נעשית בצופן בינרי — 1 מיוצג על־ידי נקודה ממוגנטת ו־0 מיוצג על־ידי נקודה שאינה ממוגנטת. התווים נכתבים על הסרט בצופן בינרי מיוחד המבוסס על 7 או 9 סיביות — כמספר הערוצים בסרט (תרשים 2.14). צפיפות רישום הנתונים על גבי הסרט הנה כ־1600 תווים באינטש.



תרשים 2.14 : סרט מגנטי

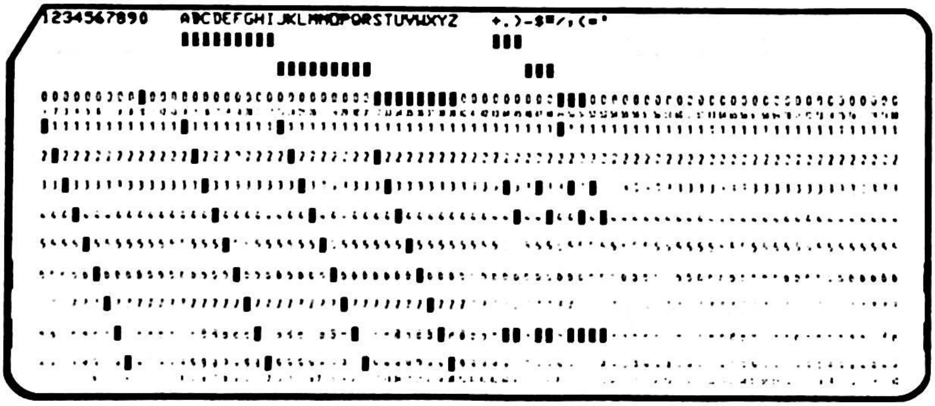
כתיבה על גבי הסרט מוחקת את הנתונים הקודמים שנכתבו על הסרט באותו המקום. יתרונותיו של הסרט המגנטי — מהירות גבוהה בקריאת נתונים ובכתיבתם, קיבולת גדולה בנפח מצומצם, מחיר זול, ואפשרות לכתוב ולקרוא ממנו למעלה מ־50,000 פעם. חסרונו בכך שהוא דורש רישום נתונים בסדר עוקב, דבר המונע הוספת נתונים חדשים לתוך סדרת נתונים קיימת.

קַסֶטָה (cassette tape)

הקסטה היא גירסה מוקטנת של הסרט המגנטי ודומה מאד לקסטה הנמצאת כיום בשימוש להקלטה ולשידור מוסיקה. יתרונה במחירה הזול יחסית.

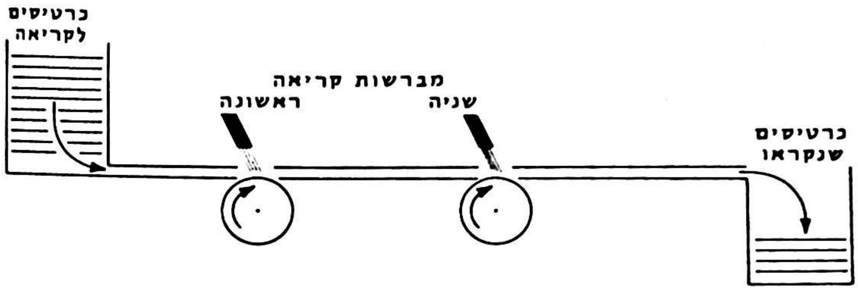
קוראת ומנקבת כרטיסים (card reader and punch)

הפעולה נעשית באמצעות כרטיסי קרטון מנוקבים. הכרטיס המקובל מחולק ל־80 טורים ול־12 שורות בכל טור (תרשים 2.15). שתים־עשרה השורות משמשות לציון התו המתאים על־ידי ניקובן בצופן מוסכם. ניתן לפענח את התו המנוקב בכל טור ולהדפיס אותו בראש הכרטיס. הפינה השמאלית־עליונה של הכרטיס חתוכה, כדי לוודא שהוא מונח בצורה הנכונה בזמן הקריאה.



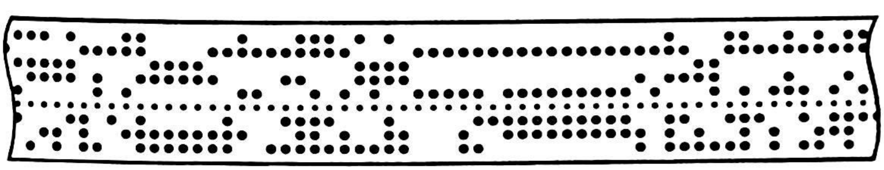
תרשים 2.15 : כרטיס מנוקב

הציוד לקריאת כרטיסים מסוגל לקרוא בין 500 ל־2000 כרטיסים בדקה. בקוראת הכרטיסים הכרטיס עובר בין גלגל נע למברשת המבחינה בין השורות המנוקבות לבלתי מנוקבות בכל טור, ומעבירה את התווים המוצפנים הללו ליחידת העיבוד המרכזית באמצעות אותות חשמליים. קיימות גם קוראות כרטיסים המבחינות בניקוב באמצעות תא פוטו־אלקטרי רגיש לאור במקום מברשת. (ראה תרשים 2.16).



תרשים 2.16: קוראת כרטיסים

סרט נייר מנוקב (pnched paper tape)
הנתונים מנוקבים על גבי סרט נייר בדומה לסימנים המגנטיים שעל הסרט המגנטי. כל נקודה לאורך סרט הנייר יכולה להיות מנוקבת או לא מנוקבת. מצבים המבטאים את הסיביות 0 ו־1. כל תו מבוטא בצופן בינרי של 7 סיביות (תרשים 2.17). יתרונותיה של שיטה זו במחירה הנמוך; חסרונותיה — זמן קריאה וכתיבה ארוך מאוד, שימוש חד־פעמי בסרט הנייר, ואי־אפשרות לשנות או להוסיף נתונים לאלו הנמצאים כבר על סרט הנייר.

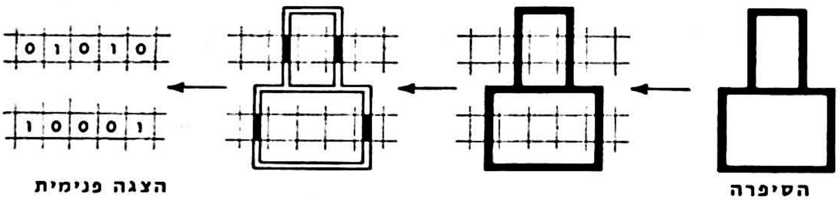


תרשים 2.17: סרט נייר מנוקב

יחידות לקריאה אופטית או מגנטית
קיימות מספר שיטות המאפשרות קריאת נתונים ישירות ממסמכים, דבר החוסך את שלב הכנת הנתונים למחשב.

קריאת סימנים (mark reading) — בשיטה זו הטפסים המיועדים לקריאה מחולקים למשבצות, וממלא הטופס מסמן בעפרון או בעט רגיל את המשבצות המתאימות לתווים הרצוים לו בכל שורה ועמודה. הטופס המסומן עובר דרך היחידה לקריאת סימנים, הסורקת את הטופס ומתרגמת את המשבצות הריקות והמסומנות לאותות חשמליים המועברים אל המחשב.

קריאת תווים אופטית (optical character recognition) — בשיטה זו נעשה שימוש באותיות ובספרות בעלות מבנה מוסכם וקבוע לצורך רישום הנתונים על הטפסים. הטופס עובר דרך היחידה לקריאת תווים, הקוראת כל תו בצורה אופטית, מחלקת את ה"תמונה" שהתקבלה למשבצות או לנקודות, ומתרגמת את המשבצות הבהירות והכהות לאותות חשמליים (תרשים 2.18).



תרשים 2.18: קריאה אופטית של הספרה 8

קריאת תווים מגנטית (magnetic ink character recognition) — בשיטה זו התווים נרשמים על הטפסים בדיו מגנטית. הטופס מועבר דרך היחידה לקריאת תווים מגנטיים, המקבלת זרם חשמלי היחסי לכמות הדיו המגנטית בכל תו. יתרונה של השיטה ביכולת לזהות את התווים שנכתבו בדיו מגנטית, גם אם הודפסו עליהם תווים אחרים (למשל — חותמת בנק המוטבעת על המחאה שנכתבה בדיו מגנטית). שיטה זו מקובלת כיום ברוב הבנקים לסימון המחאות.

קריאת קווים אופטית (bar code) — שיטה זו מקובלת לסימון מוצרים במחסנים ובנקודות מכירה. בשיטה זו, צירוף של קווים בעלי עוביים שונים מסמן קוד מוסכם. הקווים נקראים באמצעי חישה המחובר למחשב, שבו מפוענח הקוד והופך לנתון (ראה תרשים 2.19).

0 71610 61002

תרשים 2.19: קוד קווים

יחידות להכנת סרטים מיקרופילם

יחידות אלה מעבירות את הנתונים המתקבלים מן המחשב ישירות לסרטי מיקרו־פילם. התווים מסומנים על גבי סרט המיקרופילם בצורת אותות וספרות הניתנות לקריאה בעזרת מסך הגדלה. יחידות אלה מעבירות בין 6000 ל־30,000 שורות בדקה לסרטי מיקרופילם. סרט מיקרופילם בגודל של כ־10 סנטימטרים מרובעים יכול להכיל תוכנם של כ־300 עמודי נייר מודפסים!

תוויין (graph plotter)

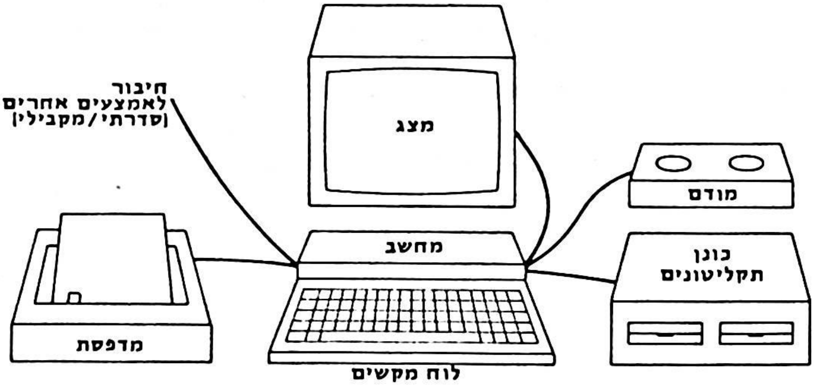
התוויין מתרגם את האותות החשמליים המתקבלים מן המחשב לשרטוט על גבי נייר. התוויין שימושי בעיקר בהכנת גרפים ותרשימים, שהם תוצאות של עיבודים במחשב.

המחשב האישי (PERSONAL COMPUTER)

המחשב האישי מאפשר כיום (במחיר של פחות מ־500 $) לכל אחד מאתנו להחזיק במחשב, בעל אותה עצמת חישוב שהיתה קיימת במחשבים הגדולים של שנות ה־60 ובמחשבים הקטנים של שנות ה־70.
לפני עשרים שנה ניתן היה להצדיק קניית מחשב רק אם ענה על דרישות ארגונים גדולים. המחשבים הזעירים אשר הופיעו בשנות ה־70 אפשרו את רכישתם עבור מחלקות או קבוצות בתוך הארגון, וכיום, ניתן גם ליחיד לרכוש מחשב אישי. בד בבד להוזלה במחיר חלה גם התקדמות טכנולוגית שאפשרה לבנות מחשבים, אשר מבחינת האינטראקציה שבין האדם למכונה הם פשוטים וקלים לתפעול, ולכן מאפשרים לאנשים ללא רקע טכני לתפעלם.
המחשב האישי הראשון הגיע לשוק ב־1975, ומאז נמכרו כבר מיליוני מחשבים כאלו. עקב הפיתוח המואץ, הן מבחינה טכנולוגית והן מבחינת השימוש הרחב במחשבים, החלו לדבר על "מהפיכת המחשבים".
מחשב אישי הוא מחשב קטן המבוסס על מיקרופרוסטור, ולכן נקרא מיקרו מחשב (microcomputer).
מיקרו מחשב מסוגל לטפל בפעילות אחת, כגון פיקוח על מכונות יצור, או פיקוח על זרימת הדלק למנוע של מכונית; הוא יכול לשמש מעבד תמלילים, משחק אלקטרוני (video game), או מחשב כיס. המחשב האישי שונה במקצת; הוא מחשב עצמאי אשר מעמיד לרשות המשתמש מספר רב של שימושים אפשריים. מקובל להגדיר כמחשב אישי מחשב שיש לו את התכונות הבאות:

— מחיר מערכת מחשב שלמה הוא פחות מ־5,000 $.
— המחשב כולל או ניתן לחיבור לזיכרון משני, כגון קסטה או דיסקט.
— המיקרופרוססור יכול לתמוך בזיכרון של 64KB או יותר.
— המחשב מסוגל לפעול בעזרת שפת תכנות עילית אחת לפחות.
— מערכת ההפעלה מאפשרת דו־שיח אדם/מחשב; המחשב מגיב מיד לדרישות המשתמש.
— משווק על־ידי צינורות שיווק המוניים, עם דגש על מכירה למחוסרי ניסיון, אשר לא השתמשו במחשב קודם לכן.
— המערכת גמישה ומאפשרת לבצע מספר רב של תכניות בנושאים שונים; המחשב אינו מיוצר עבור יישום ייחודי.

מערך אבזורי החומרה של מחשב אישי (תרשים 2.20) כולל יחידות המאפשרות עיבוד ואחסון של נתונים, וכן אפשרות תקשורת עם יחידות אחרות. ערוץ תקשורת מהיר מקשר בין היחידות העיקריות של המחשב האישי.
ניתן להזין את המחשב בנתונים באמצעות לוח המקשים או באמצעות דיסקט. הזיכרון הראשי מאחסן בתוכו תכניות ונתונים. הדיסקט מאפשר אחסון כמויות גדולות של נתונים יותר מבזכרון הפנימי, אולם הוא אטי ממנו. הממשקים למיניהם מאפשרים חיבור של יחידות נוספות, כמו מדפסת, או התקשרות למחשב אחר באמצעות קו טלפון או כל קו תקשורת אחר. בממשק סריאלי מועבר המידע ברצף, סיבית אחר סיבית, בעוד שבאמצעות ממשק מקביל מועברות מספר סיביות בו־זמנית.



תרשים 2.20: מערך החומרה של מחשב אישי

תוכנה

מכלול התכניות במערכת ממוחשבת נקרא תוכנה. התוכנה מאפשרת לחומרה (מכלול פריטי הציוד) לפעול ולבצע את העיבודים הנדרשים.

ללא תכנה משולה החומרה למכונית ללא נהג; לפטיפון ללא תקליטים.
תכניות המחשב מתחלקות לשתי קבוצות עיקריות — תוכנה בסיסית ותוכנה יישומית.

התוכנה הבסיסית היא אוסף תכניות המחשב המפעילות את המחשב עצמו, ומאפשרות הפעלה של תכניות אחרות — חיצוניות. התכנה הבסיסית מותאמת בדרך־כלל למחשב מסוים ונמכרת יחד אתו.

התוכנה היישומית היא אוסף תכניות מחשב המבצעות את היישום המסוים הנדרש — ניהול מלאי בארגון או משחק. התוכנה היישומית נרכשת או מפותחת על־ידי המשתמש עצמו, בהתאם לצרכיו.

תכניות מחשב

תכנית מחשב היא סדרת הוראות ברורות וחד־משמעיות, שיש סדר לוגי ביניהן. הן המבצעות את הפעולות על נתונים, והן הקובעות את דרך הפעולה של המחשב ואת תוצאות הפעולה.
כדי להכין תכנית מחשב, חייב כותב התכנית להבין את שלבי העבודה הנדרשים ואת מהות הנתונים שעליהם נעשות הפעולות, עליו לנתח את כל המצבים האפשריים במהלך ביצוע העבודה, ולאחר מכן לתכנן את שלבי הביצוע במחשב ולתרגמם להוראות בשפת תכנות.

שפת התכנות (programming language) היא אוסף מוגבל של מלים וכללי תחביר אשר המחשב מסוגל ל"הבין". מלים אלה הן הוראות (instructions). בנוסף להן נעשה שימוש גם במשתנים (variables) המקבלים ערכים מנתונים (data) שעליהם מבוצעות ההוראות. באמצעות ההוראות והמשתנים, ולפי כללי התחביר (syntax), כותב התכניתן משפטים (statements), המנחים את המחשב.

שפת התכנות מורכבת בדרך כלל ממלים (או מקיצורי מלים) מן השפה האנגלית, ולכן קל לתכניתן לכתוב את התכנית. לאחר הזנת התכנית למחשב עוברת התכנית בדיקה ותרגום (הידור) על ידי תכנית מתרגמת (מהדר — compiler), הבודקת את ההוראות והופכת אותן משפת התכנות ה"אנושית" (שפה עילית) לשפת מחשב בסיסית. שפה זו (שפת מכונה — machine language) בנויה מאותות בינאריים, שאותם מסוגל המחשב ל"הבין" ולבצע.
בכתיבת התכנית בשפת התכנות יש להקפיד על כתיבה מדויקת של המשפטים בהתאם לכללי השפה (כתיב ותחביר), ולא לא תוכל התכנית המתרגמת לתרגם את ההוראות לשפת מכונה. התכנית המתרגמת מבצעת תרגום טכני לחלוטין. היא איננה מסוגלת להבין את כוונת התכניתן כאשר היא נתקלת במשפט בלתי שלם, בלתי ברור או שגוי, בשגיאת כתיב או במלה לא מוכרת, ומשום כך אינה מסוגלת לתקן שגיאות אלא רק להתריע עליהן.

דוגמה:

כדי להבין מהי תכנית מחשב נתייחס לדוגמה הבאה:
מפעל מייצר עשרה סוגים של מוצרי חשמל. המפעל מוכר את המוצרים לחנויות באמצעות חמישה סוכני מכירה, המקבלים עמלות לפי היקף המכירות.
עמלת סוכן נקבעת לפי שיעור העמלה של כל סוג מוצר.

מנהל המפעל ביקש מפקיד המכירות לחשב את העמלות של סוכני המכירה לפי מכירות שנעשו בחודש החולף. כעבור זמן־מה הביא פקיד המכירות רשימה של סוכני המכירה ושל העמלות המגיעות לכל אחד, ומנהל המפעל קיבל את מבוקשו.

מה למעשה קרה ?

ננתח את פעולותיו של פקיד המכירות מרגע קבלת הבקשה ועד שהניח את הרשימה בפני המנהל :

1. הבין את בקשת המנהל.
2. הכין טבלה, ובה שורה לכל סוכן ועמודה לכל סוג מוצר.
3. לקח את קלסר החשבוניות שהוצאו בחודש הקודם.
4. עבר על החשבוניות, ועבור כל סוג מוצר הוסיף את ערך המכירות לתא המתאים בטבלה (סוכן/סוג מוצר).
5. שלף את טבלת העמלות למוצרי החברה.
6. בכל תא בטבלה חישב את סכום העמלה המגיע לסוכן (ערך מכירות כפול שיעור העמלה).
7. סיכם את סכומי העמלה לכל סוכן לכל סוגי המוצרים שמכר, והגיע לתוצאה הרצויה.
8. הדפיס את הרשימה והגיש אותה למנהל המפעל.

מן הדוגמה לעיל נראה כי "בקשה" פשוטה של מנהל המפעל "תורגמה" לתהליך רב־שלבי. מאחורי תיאור השלבים בדוגמה מסתתרות מספר הנחות: פקיד המכירות יודע מי הם סוכני המכירות (כלומר — בידיו רשימה) ; יש לו גישה לחשבוניות שהוצאו בחודש החולף ; ברשותו טבלת שיעורי עמלות לפי סוגי המוצרים.

אילו היה פקיד המכירות המנוסה נעדר מעבודתו, ומחליפו היה פקיד לא מנוסה — לא היה הלז מצליח "להבין" את בקשת המנהל ולהשיג את התוצאה הדרושה.

הערה :
מטרת הדוגמה להמחיש את העקרונות ; אפשר לרדת לפירוט יתר של שלבי הביצוע, ואפשר, כמובן, לבצע אותה עבודה בצורה ו/או בסדר אחר.

קיים דמיון רב בין סדרת פעולותיו של פקיד המכירות לבין תכנית מחשב. "בקשה" אחת של מנהל המפעל גרמה לביצוע סדרת פעולות ולהפקת התוצאה המבוקשת.

שלבי הכנת תכנית מחשב

הכנת תכנית מחשב היא תהליך טכני בעיקרו, והיא כרוכה בביצוע מספר שלבים הכרחיים. כל השלבים חייבים להיעשות בסדר לוגי קבוע; ויתור על שלב כלשהו יגרום בהכרח לתקלות בשלבים העוקבים.

שלבי הכנת התכנית:

1. בחינה והבנה של מטרת התכנית ומרכיביה
בשלב זה נקבעים המושגים והדרישות הכלליות מהתכנית. נוצר הבסיס להגדרת התכנית על ידי מתכנן התכנית, באמצעות הבנת הצרכים והדרישות של המשתמש בתכנית.
2. קביעת המסגרת והגבולות של התכנית — התחלה, סיום, קלטים, פלטים, נוסחאות חישוב, וכדומה
בשלב זה צריך מתכנן התכנית להבין מתי תופעל התכנית, איך תתחיל וכיצד תסיים את עבודתה, מהם הנתונים שאותם צריכה התכנית לקבל על מנת שתוכל לפעול, מהיכן תקבל נתונים אלה, מהן נוסחאות החישוב, מהם השלבים הלוגיים של ביצוע העבודה, ומהי התוצאה הנדרשת מהפעלת התכנית.
3. הגדרת הנתונים ומאגרי הנתונים שבהם עושה התכנית שימוש
כל הנתונים הנדרשים על־ידי התכנית מוגדרים בשלב זה, ומאופיינים כנתוני קלט, נתוני ביניים ונתוני פלט. מאגרי הנתונים מוגדרים מבחינת מבנה המאגר, צורת הגישה לנתונים במאגר, מבנה הרשומה ומהות הנתונים ברשומה.
4. תכנון שלבי עבודת התכנית ואפיון כל האפשרויות
מטרת שלב זה לקבוע את כל המהלכים שתבצע התכנית ואת מהלך הזרימה הלוגי הקושר מהלכים אלה. הכלי המקובל לשימוש בשלב זה הוא תרשים זרימה של התכנית, המתאר בפירוט את כל המהלכים שתבצע התכנית ואת מהלך הזרימה ביניהם.
5. תכנות התכנית והזנתה למחשב
שלב התכנות הוא שלב טכני בעיקרו — מעבר מהתכנון המפורט של התכנית לתכנית כתובה בשפת תכנות. בשלב זה נעשה, אם כן, "תרגום" כל שלב בתכנון לקבוצת פקודות בשפת התכנות.
בגמר כתיבת התכנית בשפת התכנות מוזנת התכנית למחשב (במישרין — על־ידי הקשתה במסוף המחשב, או בעקיפין — הקשתה על גבי דיסקט או אמצעי אחר). בשלב זה התכנית נמצאת במחשב וניתנת לתרגום לשפת מכונה ולהפעלה לאחר התרגום. לתכנות בשפה עילית מסוימת נתייחס בפרק 3.

6. בדיקה, ניפוי וניסוי של התכנית

שלב זה מתחלק לשני תת־שלבים: א. בדיקה ותרגום על־ידי המהדר, ב. בדיקה וניסוי על־ידי מתכנן/מתכנת התכנית.

בדיקה על ידי המהדר — התכנית עוברת בדיקה ותרגום על־ידי תכנית מחשב אחרת (המהדר — compiler). המהדר בודק את התכנית מבחינת הכתיב והמבנה התחבירי של ההוראות. אם המהדר נתקל בהוראה "שאיננו מבין", הוא מתריע על השגיאה (bug) ומצביע על ההוראה השגויה, אך איננו מתקן את השגיאה. התכניתן עובר על תוצאות מהלך ההידור ומבצע את התיקונים הדרושים (מבצע ניפוי — debugging). לאחר מכן הוא מחזיר את התכנית למהדר לשם בדיקה נוספת. בגמר שלב זה נקייה תכנית המחשב משגיאות הידור (אך עדיין יתכנו שגיאות אחרות בתכנית!). התכנית הנקייה משגיאות מתורגמת על ידי המהדר לתכנית בשפת מכונה, שניתן להפעיל אותה במחשב.

ניסוי על ידי מתכנן/מתכנת התכנית — בשלב זה התכנית מופעלת במחשב. הפעלת התכנית במחשב משמעותה שהמחשב מבצע את ההוראות כפי שנכתבו בתכנית; כדי להפעיל את התכנית במחשב יש לספק לה את הנתונים הנדרשים על ידה. בשלב הניסוי מזין התכניתן את התכנית בנתוני הניסוי (test data) שעליהם מבצעת התכנית את הפעולות. התכניתן בודק את כל האופציות, וכן אם התכנית מפיקה את התוצאות הרצויות בצורה נכונה. אם מתגלות שגיאות בתוצאות או אם לא מושגות התוצאות הרצויות — התכניתן מתקן את התכנית, ומחזיר את התכנית המתוקנת לשלב הבדיקה על ידי המהדר. בשלב הניסוי מתבצע, למעשה, ניפוי של שגיאות לוגיות.

כאשר התכניתן משוכנע בתקינות התכנית — מוכנה התכנית להפעלה שוטפת על ידי המשתמש בתכנית.

גם בתום שלב זה ייתכנו שגיאות (bugs) בתכנית, שלא נתגלו בשלבי הבדיקה והניסוי. שגיאות אלה עלולות להופיע רק בעת ההפעלה השוטפת של התכנית.

7. תיעוד התכנית

התיעוד מורכב מריכוז כל המידע הרלוונטי להבנת התכנית. הסעיפים העיקריים בתיעוד התכנית חופפים למעשה את שלבי הכנת התכנית וכוללים:

— הסבר מטרות התכנית
— מסגרות וגבולות התכנית
— הגדרת הנתונים ומאגרי הנתונים
— תכנון התכנית
— תדפיס התכנית בשפת תכנות עילית
— תוצאות הבדיקה והניסוי (דוגמת קלטים ופלטים)

— הוראות להפעלת התכנית.

תיעוד התכנית הוא בעל חשיבות רבה, מהסיבות האלה:

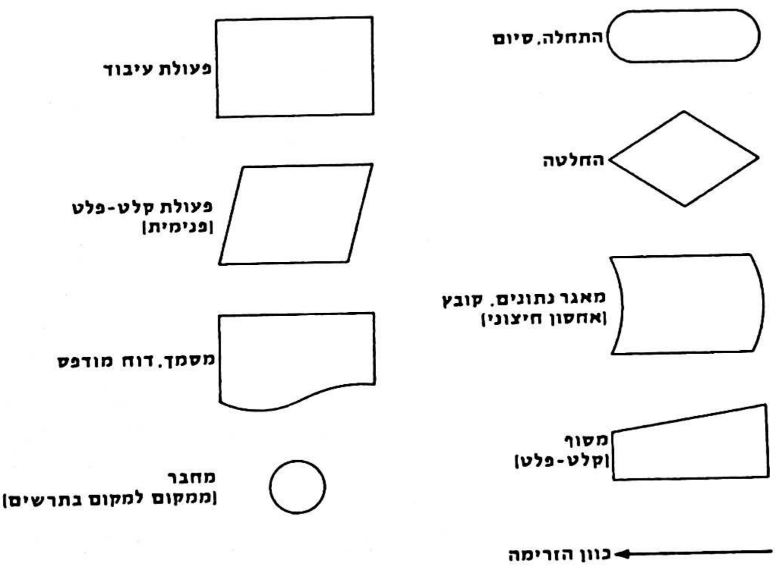
— אין ביטחון שמפתחי התכנית (המתכנן/תכניתן) ימשיכו לאחזק אותה בעתיד, ויתכן שיהיו אלה אנשים אחרים. התיעוד יאפשר לאחרים ל"היכנס" לתכנית, להבין איך בנויה התכנית ולתחזק אותה (לבצע תיקונים, שינויים ושיפורים העשויים להידרש בעתיד).
— התיעוד מאפשר לשלב את התכנית במערכת המורכבת ממספר גדול של תכניות אחרות, ושבה לכל תכנית מטלה מסוימת.
— התיעוד, הכולל גם הנחיות למשתמש, מאפשר לאנשים שלא כתבו את התכנית להשתמש בה, להפעיל אותה ביעילות, ולמצוא תשובות לשאלות העשויות להתעורר במהלך הפעלתה של התכנית.

8. שימוש בתכנית

שלב זה הוא השלב שלשמו פותחה התכנית. בשלב זה עושה המשתמש שימוש בתכנית המחשב, וזו מעבדת את הנתונים ומפיקה את התוצאות הדרושות.

תרשים תהליך, תרשימי זרימה

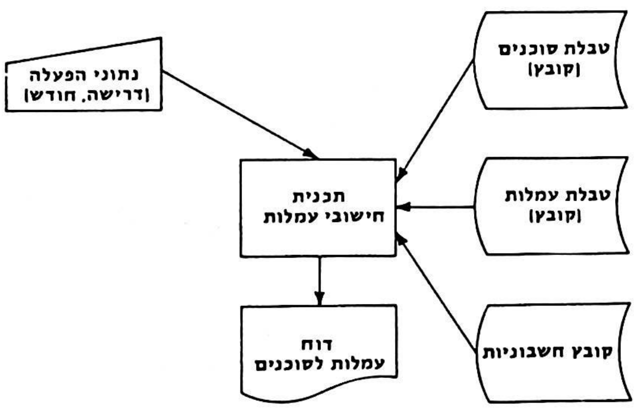
תרשים תהליך ותרשים זרימה הם התיאור הגרפי של תהליך ביצוע ההוראות בתכנית; כל הוראה או קבוצת הוראות מיוצגות על־ידי תבנית מוסכמת. התבניות המקובלות להכנת תרשימים:



ההבדל העיקרי בין תרשים תהליך (process flow chart או block diagram) לתרשים זרימה (program flow chart או flow chart) הוא ברמת הפירוט.

בתרשים תהליך מייצגת תבנית העיבוד מהלך עיבוד שלם (תכנית שלמה או קטע תכנית); בתרשים זרימה מייצגת תבנית העיבוד פקודה אחת או קבוצת פקודות מצומצמת. האבחנה איננה חד־משמעית, והאפיון נעשה לפי צורת השימוש ורמת הפירוט הנדרשת.

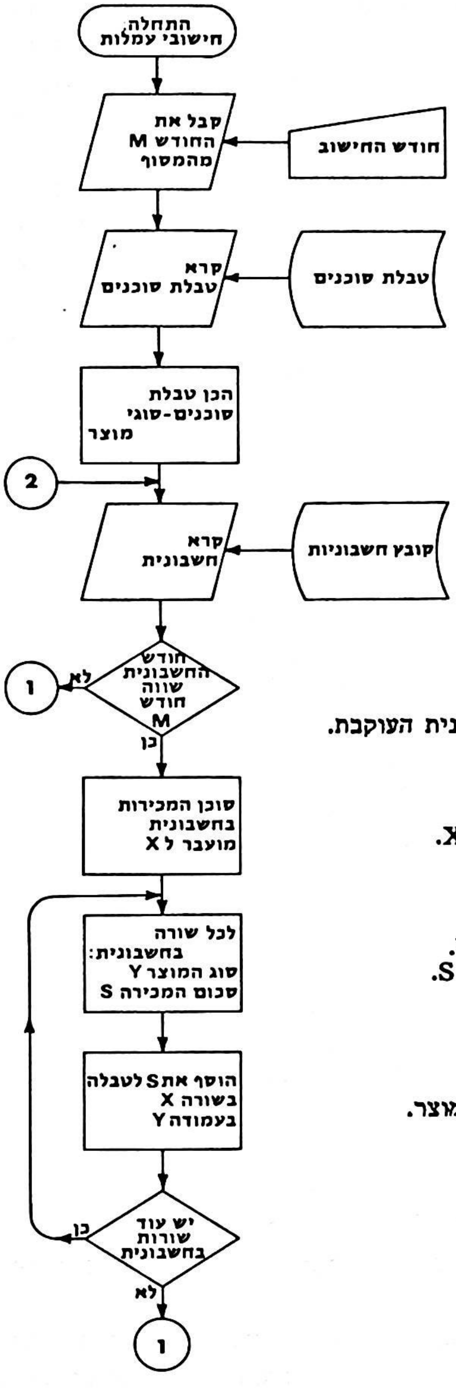
השימוש בתרשים תהליך ובתרשים הזרימה יומחש באמצעות הדוגמה לחישוב העמלות לסוכנים (ראה תרשים 2.21).



תרשים 2.21: דוגמה לתרשים תהליך

התרשים מראה את הקלטים, הפלטים ומאגרי הנתונים (קבצים) שבהם עושה התכנית שימוש, מבלי לפרט את שלבי ביצוע הפעולות בתכנית.

תרשים הזרימה המוצג להלן הנו ברמת פירוט גבוהה מאוד, ומראה את כל הצעדים שהתכנית תבצע.



תבנית התחלה:
התחלת תכנית חישוב עמלות

תבנית קלט ממסוף:
חודש החישוב M הוא החודש עבורו נדרש הדוח. M יהיה מספר בין 1 ל־12.

תבנית קלט ממאגר נתונים (טבלה):
טבלת הסוכנים כוללת את מספרו הסדורי של כל אחד מחמישה סוכני מכירה.

תבנית עיבוד:
טבלת סוכנים/סוגי מוצר כוללת שורה לכל סוכן, ועמודה לכל אחד מעשרה סוגי מוצר. בשלב ההכנה הטבלה ריקה.

תבנית קלט ממאגר נתונים (קובץ):
קובץ החשבוניות מכיל את נתוני החשבוניות, הכוללים: חודש החשבוניות, מספר הסוכן, ושורה לכל מוצר שנמכר, הכוללת: סוג המוצר, סכום המכירה.

תבנית החלטה:
בדיקה אם החשבונית שנקראה שייכת לחודש החישוב. אם לא — ממשיכה התכנית לתבנית המסומנת ב־1. אם כן — ממשיכה התכנית לתכנית העוקבת.

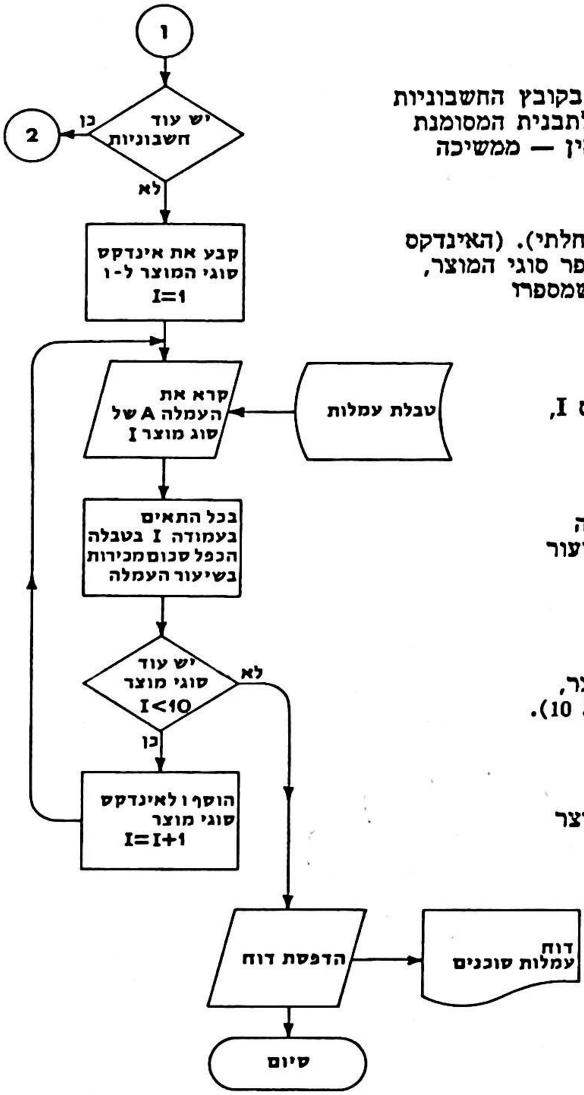
תבנית עיבוד:
התכנית מזהה את מספר סוכן המכירות לפי החשבונית, ומעבירה את המספר למשתנה X.

תבנית עיבוד:
התכנית מזהה לפי השורה בחשבונית את:
א. סוג המוצר, ומעבירה את המספר למשתנה Y.
ב. סכום המכירה, ומעבירה את הסכום למשתנה S.

תבנית עיבוד:
התכנית מוסיפה את הערך של המשתנה S (סכום המכירה) לתא בשורה X (הסוכן) בעמודה Y (סוג המוצר) בטבלת הסוכנים/סוגי מוצר.

תבנית החלטה:
התכנית בודקת אם יש שורות נוספות בחשבונית. אם יש — חוזרת לתבנית המטפלת בשורה בחשבונית; אם אין — ממשיכה התכנית לתבנית המסומנת ב־1.

תבנית מחבר:



תבנית החלטה:
התכנית בודקת אם יש עוד חשבוניות בקובץ החשבוניות שלא נקראו. אם יש — חוזרת התכנית לתבנית המסומנת ב־2 (לקריאת חשבונית מהקובץ). אם אין — ממשיכה התכנית לתבנית העוקבת.

תבנית עיבוד:
אינדקס סוגי מוצר נקבע ל־1, (ערך התחלתי). (האינדקס ישונה על־ידי התכנית ל־2, 3, . . . כמספר סוגי המוצר, ובכל פעם תטפל התכנית בסוג המוצר שמספרו כמספרו של האינדקס).

תבנית קלט ממאגר נתונים (טבלה):
התכנית תקרא מהטבלה את העמלה של סוג מוצר שמספרו כמספרו של האינדקס I, ותעביר את שיעור העמלה למשתנה A.

תבנית עיבוד:
התכנית עוברת על כל התאים בעמודה שמספרה כמספרו של אינדקס I ומכפילה את סכום המכירות הרשום בכל תא בשיעור העמלה A, ואת התוצאה תשמור באותו תא בטבלה.

תבנית החלטה:
התכנית בודקת אם טופלו כל סוגי המוצר, (כלומר אם אינדקס סוגי המוצרים שווה 10).

תבנית עיבוד:
התכנית מוסיפה 1 לאינדקס סוגי מוצר (כלומר התכנית עוברת לטפל בסוג המוצר הבא).

תבנית פלט — דוח מודפס:
התכנית מדפיסה דוח עמלות הסוכנים, מתוך הטבלה שנבנתה.

תבנית סיום:
סיום התכנית לחישובי עמלות.

המעבר מתרשים הזרימה דלעיל לתכנית מחשב הוא כמעט מיידי: כל תבנית מתורגמת להוראה או לקבוצת הוראות בשפת התכנות.

פרק 3.

תכנות

תכנות של תכנית מחשב פירושו שהתכנון המפורט של התכנית (שהוכן כתרשים זרימה או בכל צורה אחרת) נכתב בשפת התכנות.
שפות התכנות בנויות מאוסף מצומצם של מלים בשפה האנגלית ומכללי תחביר. הדמיון לשפה האנגלית מפשט את כתיבת התכנית עבור התכניתן; שפת התכנות נקראת "שפה עילית" — קל לאדם להבין אותה. בשלב שהתכנית מועברת למחשב, היא עוברת תרגום לשפת מכונה — שאותה קשה מאוד לאדם להבין.
שפת המכונה (machine language) נקראת גם שפה נמוכה (low level language), והיא מבוטאת באמצעות אותות בינאריים; לעומתה נקראת שפת התכנות שבה משתמש התכניתן שפה עילית (high level language), והיא מבוטאת באמצעות מלים המבוססות על השפה האנגלית, ספרות וסימנים מקובלים.

שפות התכנות הנפוצות הן:

COBOL — COmmon Business Oriented Language שפה אשר תוכננה במיוחד לכתיבת תכניות בשטח המנהלי־עסקי.

FORTRAN — FORmula TRANslator — שפה המשמשת בעיקר לכתיבת תכניות ליישומים מדעיים.

PL/1 — Programming Language /1 — שפה המבוססת על שילוב התכונות החזקות של COBOL ו־FORTRAN.

BASIC — Beginners All-purpose Symbolic Instruction Code — מתאימה לכתיבת תכניות בכל שטח. ייחודה בפשטות הלימוד והשימוש.

קיימות שפות תכנות רבות נוספות; כל שפה מאופיינת בתכונות המיוחדות לה. בספר זה בחרנו להציג את שפת ה־BASIC, עקב פשטות הלימוד והכתיבה של שפה זו, היותה אינטראקטיבית, וקיומה ברוב סוגי המחשבים — בעיקר האישיים. דיוננו איננו מיועד להכשיר תכניתנים מקצועיים; מטרתו להמחיש את עקרונות התכנות ולהקנות כלים בסיסיים לכתיבת תכניות פשוטות.

שפת התכנות BASIC

הערה: שפת ה־BASIC מתוארת להלן בצורה בסיסית. עשויים לחול שינויים הן בצורת הכתיבה והן בכללי הכתיבה, בהתאם לסוג המחשב העומד לרשותך: על־כן, עיין גם בחוברת ה־BASIC של המחשב.

כבכל שפת תכנות עילית, גם לשפת ה־BASIC כללים קבועים של סדר וצורה, שעל פיהם נכתבות ההוראות. השימוש בשפת תכנות זאת הוא בעיקר בעזרת מסופים הקשורים למחשב, כך שניתן להזין את התכנית למחשב ולבצעה מיד. התכנית (program) בשפה זו מורכבת משורות (lines) שלכל אחת מהן מספר (line number). בכל שורה תופיע הוראה (statement) אחת. ביצוע ההוראות על־ידי המחשב נעשה לפי סדר עולה של מספרי השורות, אלא אם כן קיימת הוראה לעבור למקום אחר בתכנית. כל הוראה כוללת מלת מפתח (key-word), המזהה את ההוראה.

הכללים הבסיסיים של שפת ה־BASIC:

1. הוראות התכנית כתובות רק באותיות גדולות (CAPITAL LETTERS).
2. כל שורה (הוראה) מתחילה עם מספר, הנקרא מספר שורה או מספר הוראה. לפני ביצוע התכנית, מסדר המחשב את ההוראות בסדר עולה לפי מספרי השורה ומאחסן אותן בזיכרון. מספרי השורות משמשים ככתובות להוראות בתכנית. ביצוע התכנית מתחיל מההוראה הראשונה וממשיך לפי סדר ההוראות ובהתאם להוראות ההסתעפות בתכנית.
3. בכל שורה מלת מפתח (key-word), המגדירה את סוג ההוראה.

מושגים והגדרות

תו (character) הוא המרכיב הלוגי הקטן ביותר הקיים בשפה העילית. כל שורה בתכנית מורכבת מתווים.

ניתן לסווג את התווים לשלוש קבוצות:

א. תו אלפביתי — (alphabetic character) A — Z

ב. תו מספרי — (numeric character) 0 — 9

התווים הללו יחד מכונים אלפאנומרים (alphanumeric).

ג. תווים מיוחדים — (special characters)

תו הרווח (space), פסיק, נקודה, סימנים אלגבריים וכד׳.

קבוע מספרי — (numerical constant)

קבוע מספרי מייצג ערכים מספריים שאינם משתנים בזמן ביצוע התכנית.
קבוע מספרי נכתב כסדרת תווים מספריים (אחד או יותר), ויכול להכיל סימן (+ או —) ונקודה עשרונית.
קבוע מספרי שלם נכתב ללא נקודה עשרונית 4 23
קבוע מספרי עשרוני נכתב עם נקודה עשרונית 7.1 0.25 0.5
ניתן לכתוב ב-BASIC קבוע מספרי בצורה מעריכית (exponential).
למשל: 5 — 2.7E שפירושו 2.7 — כפול 10^5, או 0.00027—.
בשפת ה-BAISC מודפסים מספרים המכילים עד 6 ספרות משמעותיות בצורת מספר שלם או עשרוני. ולא, יודפס המספר בצורה מעריכית מנורמלת ויעוגל ל־6 ספרות משמעותיות. למשל, המספר 0.00000187 יודפס בצורה $1.87E^-6$.
הצורה המנורמלת היא כאשר הנקודה העשרונית ממוקמת מימין לספרה המשמעותית הראשונה.

משתנה מספרי (numerical variable)

משתנה מספרי מייצג ערך מספרי שיכול להשתנות בזמן ביצוע התכנית.
משתנה מספרי מיוצג על־ידי תו אלפביתי בודד או תו אלפביתי ואחריו תו נומרי, כגון: A, Q1, Z.
ערך המשתנה יכול להשתנות בזמן ביצוע התכנית על־ידי הוראות התכנית.
כל משתנה מספרי, שערכו לא נקבע על־ידי הוראה מפורשת בתכנית, מקבל בשפת ה-BASIC את הערך ההתחלתי 0.

אופרטור אריתמטי (arithmetical operator)

אופרטורים אריתמטיים משמשים לצד קבועים מספריים, משתנים מספריים וסימני הסוגרים להצגת ביטויים אריתמטיים. האופרטורים האריתמטיים מבוטאים באמצעות הסמלים האלה:

הסמל	הפעולה	דוגמה	הסבר
+	חיבור	A + B	חבר B ל־A
—	חיסור	A — B	חסר B מ־A
*	כפל	A * B	כפול B ב־A
/	חילוק	A / B	חלק A ב־B
↑ או **	חזקה	A ↑ 2	העלה A בחזקת 2

ביטוי אריתמטי (arithmetical expression)

ביטוי אריתמטי הוא קבוע מספרי בודד, או משתנה מספרי בודד וכן כל צירוף אחר מותר שלהם הנעשה בעזרת האופרטורים הארתמטיים והסוגריים.

למשל:

$$(A \uparrow B + B \uparrow 2) \ / \ (3*X)$$

סימן המינוס (מינוס אונארי) יכול לבוא לפני כל ביטוי אריתמטי ואף אחרי אופרטור אריתמטי, ללא צורך בסוגריים. הסוגריים המותרים לשימוש הם רק הסוגריים העגולים ( ), המשמשים לקביעת סדר ביצוע הפעולות.

המחשב קובע את ערכם של הביטויים האריתמטיים לפי ערך הקבועים והמשתנים המספריים המופיעים בהם ברגע חישוב הביטוי. חישוב הביטוי האריתמטי נעשה משמאל לימין, חוץ מבאופרטורים בעלי עדיפות גבוהה, המבוצעים לפני אופרטורים בעלי עדיפות נמוכה. סדר העדיפות בין האופרטורים הוא (מהגבוה לנמוך):

1. סוגריים
2. חזקה, מינוס אונארי
3. כפל, חילוק
4. חיבור, חיסור

דוגמאות:

א. 2**4/2+3 יתבצע כך: קודם תתבצע החזקה 2^2 שתוצאתה 4. לאחר מכן תתבצע החלוקה 4/4 שתוצאתה 1, ואז יבוצע החיבור 1+3 והתוצאה הסופית תהיה 4.

ב. 2**(4/2)+3 יתבצע כך: קודם יחושב הביטוי בסוגריים 4/2 שתוצאתו 2. לאחר מכן תתבצע החזקה 2^2 שתוצאתה 4, ואז יבוצע החיבור 4+3 והתוצאה הסופית היא 7.

ג. 2**(4+3)/2 יתבצע כך: קודם יחושב הביטוי בסוגריים 4+3 שתוצאתו 7, לאחר מכן תתבצע החזקה 2^2, ואז יתבצע החילוק 7/4 והתוצאה הסופית היא 1.75.

ד. 2**(4/2+3) יתבצע כך: קודם יחושב הביטוי בסוגריים: ראשית תבוצע החלוקה 4/2 שתוצאתה 2, לאחר מכן יתבצע החיבור 2+3 שתוצאתו 5, ולבסוף תתבצע החזקה 5^2 והתוצאה הסופית היא 25.

ה. 2**((4+3)/2) יתבצע כך: קודם יחושב הביטוי בסוגריים הפנימיים 4+3 שתוצאתו 7, לאחר מכן יחושב הביטוי בסוגריים החיצוניים 7/2 שתוצאתו היא 3.5, ואז תתבצע החזקה 3.5^2 והתוצאה הסופית היא 12.25.

הצבה (השמה) (assignment)
סימן השוויון (=) משמש להשמה של ערך הביטוי הנמצא מימינו, למשתנה הנמצא משמאלו. בניגוד לכללי האלגברה, מותר בשפות התכנות לרשום אותו משתנה משני צדי השוויון, כיון שההשמה נעשית תמיד מימין לשמאל.

דוגמאות:

A = 1 הצב את הערך 1 למשתנה A.

A = 2 + 7/3 חשב את הביטוי 2 + 7/3 והצב את התוצאה למשתנה A.

A = A * 3 חשב את הביטוי A*3 כאשר הערך של המשתנה A הוא ערכו בזמן ביצוע ההוראה והצב את התוצאה למשתנה A (כלומר, הערך החדש של A יהיה פי שלושה מערכו לפני ביצוע ההוראה).

אופרטור־יחס — (relational operator) (בקיצור (R)).
בין מספרים קיימים שבעה אופרטורי־יחס הנכתבים כדלקמן:

הסמל	היחס	דוגמה	הסבר
=	שווה	A = B	A שווה ל־B
<	קטן	A < B	A קטן מ־B
>	גדול	A > B	A גדול מ־B
<>	לא שווה	A<>B	A לא שווה ל־B
>=	לא קטן	A>=B	A גדול או שווה ל־B
<=	לא גדול	A<=B	A קטן או שווה ל־B

ביטוי־יחס (relation)
נוצר על־ידי שני ביטויים אריתמטיים שביניהם אופרטור יחס. צורת ביטוי היחס היא אפוא:

ביטוי אריתמטי (R) ביטוי אריתמטי

ביטוי היחס משמש כ"תנאי" שיש לו ערך אמת או ערך שקר (true או false). למשל: 4>8 יקבל את הערך אמת ואילו 8>4 יקבל את הערך שקר.

תיאור הוראות BASIC

בהוראה יכולים להיכלל לכל היותר 72 תווים. כל הוראה ב־BASIC נכתבת כשורה אחת. לכל הוראה מספר שורה. לפני ביצוע התכנית מסודרות ההוראות

בסדר עולה לפי מספרי השורות. כדי לתאר באופן כללי הוראות BASIC ניעזר בסימון הזה:

<LN> או <LN1>, <LN2> וכו' יסמן מספר שורה (line number)

< C > או <C1>, <C2> וכו' יסמן קבוע מספרי (constant)

< V > או <V1>, <V2> וכו' יסמן משתנה מספרי (variable)

<EXP> או <EXP1>, <EXP2> וכו' יסמן ביטוי אריתמטי (expression)

בהקשת תכנית למחשב, וכן בהקשת נתונים שהתכנית מבקשת לקבל, יש ללחוץ על מקש RETURN בגמר הקשת כל שורה. לפני הקשה זו, השורה לא נקלטת במחשב.

נוהג מקובל בכתיבת תכניות הוא לספרר את השורות תוך דילוג על מספרים. שיטה זו מאפשרת להוסיף שורות לתכנית בין שורות קיימות. מומלץ להקפיד על נוהג זה.

רשימת פקודות בסיסיות של BASIC

IF-THEN	REM	LET
ON-GOTO	END	PRINT
FOR-NEXT	READ-DATA	RUN
GOSUB-RETURN	INPUT	LIST
DEF FN	GOTO	

הערה: ההוראות LIST, RUN הן הוראות מיוחדות שאינן חלק מהתכנית בשפת ה-BASIC. הוראות אלה פועלות ישירות על התכנית באמצעות מערכת ההפעלה, ולא מתוכה.

LET — הוראת ההצבה והחישוב

מבנה ההוראה:

<LN> LET <V> = <EXP>

ביצוע הוראה זו גורם לחישוב (אם דרוש) של הביטוי EXP והצבת ערכו במשתנה V. (הוראה זו אינה הוראת שוויון אלגבראית ואינה אופרטור היחס "שווה").

ניתן להשמיט את המלה LET ללא שינוי משמעות ההוראה (זו מלת המפתח היחידה הניתנת להשמטה).

ההוראה X = 3 תגרום לכך שהמשתנה המספרי X יקבל את ערך הקבוע המספרי 3.

ההוראה LET X = X + 2 תגרום להגדלת ערך המשתנה המספרי X ב־2.

דוגמאות:

```
100 LET X=X+1
259 LET W7 = (W—X4↑3)*(Z—A)/(A—B)—17
```

PRINT — הוראת פלט הגורמת להצגה במסוף. לאחר מלת המפתח תבוא רשימה של משתנים או של ביטויים אריתמטיים המופרדים על־ידי פסיקים. כמו כן יכול להופיע אחרי מלת המפתח קבוע אלפנומרי המורכב מסדרה כלשהי של תווים הסגורה בין גרשיים (").

מבנה ההוראה:

```
<LN> PRINT EXP1, EXP2 ...
```

או

```
<LN> PRINT "..." ,EXP  ,V
```

להוראת ה־PRINT ישנן כמה מטרות:

א. להציג ערכי משתנים וערכים של ביטויים אריתמטיים (תוצאות של חישובים).
ב. להציג קבועים אלפנומריים — הודעות הכלולות בתכנית.
ג. להציג שילוב של הנ"ל.
ד. לדלג על שורה במסך.

דוגמה למטרה א':

```
100 PRINT X,X**2
135 PRINT X,Y,Z,B*B—4*A*C, (A—B)**2
```

השורה הראשונה (שורה מספר 100) תגרום להצגת הערך של X, ולאחר כמה רווחים באותה שורה להצגת הערך של X^2. השורה השנייה (שורה מספר 135) תגרום להצגת חמישה מספרים, שלושה שהם הערכים של X, Y ו־Z בהתאמה, ושניים עבור תוצאת הביטויים B^2 — 4AC ו־$(A—B)^2$ (כמובן, בתנאי שלמשתנים המספריים, A, B ו־C ניתן ערך מספרי לפני פעולת ה־PRINT).

דוגמה למטרה ב':

```
100 PRINT "NO UNIQUE SOLUTION"
430 PRINT "X VALUE", "SINE", "RESOLUTION"
```

השורה הראשונה (שורה מספר 100) תגרום להצגת הכתוב בין הגרשיים ("). אותו הדבר יתבצע על־ידי ההוראה השנייה (שורה מספר 430), אולם במקרה זה יופרדו שלושת ההודעות על־ידי מספר רווחים (שים לב לסימני הפסיק המפרידים בין שלושת ההודעות).

דוגמה למטרה ג':

```
150 PRINT "THE VALUE OF X IS" ; X
315 PRINT "THE SQUARE OF" ; X ; "IS" ; X**2
```

אם נניח, לדוגמה, שהערך של המשתנה המספרי X הוא 25, אז השורה הראשונה (שורה מספר 150) תגרום להדפסת השורה הזו:

THE VALUE OF X IS 25

השורה השנייה (שורה מספר 315) תגרום להדפסת השורה הזו:

THE SQUARE OF 25 IS 625

דוגמה למטרה ד':

250 PRINT

המחשב יקדם את מקום ההצגה במסוף בשורה אחת כשתתבצע הוראה מספר 250. למעשה, תגרום הוראה זו לרווח כפול (double space).

בין הביטויים המופיעים בהוראת ה־print מפרידים שני סוגי תווים, המשמשים בעצמם מעין "הוראה":

א) תו הפסיק (,) — הוראה זו גורמת לעבור ל"שדה ההדפסה" הבא — בכל שורה ישנם 5 שדות הדפסה ברוחב של 15 תווים כל אחד.

ב) תו הנקודה־פסיק (;) — הוראה זו מתעלמת משדות ההדפסה, דבר המאפשר הדפסה של יותר מ־5 שדות בשורה.

תווים אלה יכולים להופיע בסוף הרשימה וגם בראשה. אם בסוף הרשימה לא מופיע תו כזה, מפרש זאת המחשב כהוראה לעבור לתחילת השורה הבאה. אי־לכך, ההוראה PRINT בעצמה, ללא רשימה של משתנים ו/או ביטויים, גורמת ליצירת רווח של שורה אחת.

RUN — הוראת ביצוע התכנית.
הקשת הוראה זו במסוף גורמת לתכנית הנמצאת בזיכרון המחשב להתחיל בביצוע במחשב.

מבנה ההוראה:

RUN

שים לב שבהוראה זו אין מספר שורה — הוראה זו איננה חלק מהתכנית, אלא הוראה חיצונית הגורמת להפעלת התכנית במחשב.

סיום התכנית יקרה באחד מהמקרים הבאים:

1. התכנית תתקל בהוראת עצירה או בהוראת סיום (STOP או END).
2. המחשב יבצע את כל הפקודות בתכנית.
3. המחשב יתקל בשגיאה בתכנית (הוראה שאיננו יכול לבצע).

תרגיל — דוגמה 1

תכנית לחישוב הערך העתידי P של סכום כסף נוכחי C, כעבור N שנים, המושקע לפי שער ריבית R.

הערך העתידי בשנה N מחושב לפי הנוסחה $P=C(1+R)^N$.

```
10 LET C = 10
20 LET R = .2
30 LET N = 3
40 LET P = C*(1+R)**N
50 PRINT "FOR SUM =" ; C, "INTEREST =" ; R,N; "YEARS"
60 PRINT "FUTURE VALUE IS" ; P
```

הקש את שורות התכנית למחשב. זכור ללחוץ בסוף כל שורה על מקש RETURN. ביצוע התכנית במחשב יתחיל על־ידי הקשת ההוראה RUN. יבוצעו הפעולות האלה (בתנאי שתכנית התרגום לא תמצא שגיאות בתכנית ה־BASIC):

המחשב יתחיל את ביצוע התכנית משורה מספר 10.

שורה 10 — הערך המספרי 10 יועבר למשתנה המספרי C.

שורה 20 — הערך המספרי 2. יועבר למשתנה המספרי R.

שורה 30 — הערך המספרי 3 יועבר למשתנה המספרי N.

שורה 40 — המחשב יבצע חישוב של הנוסחה, לפי הערכים המספריים שנקבעו למשתנים המספריים N, R, C.
החישוב ייעשה לפי השלבים האלה:
(1+R) התוצאה תהיה 1.2 (2. + 1)
1.2**N התוצאה תהיה 1.728 (1.2**3)
C*1.728 התוצאה תהיה 17.28 (10*1.728)
הערך המספרי 17.28 יועבר למשתנה המספרי P.

שורה 50 — המחשב יציג על המסך את הטקסט המופיע בגרשיים ואת ערכי המשתנים המופיעים בשורה.
השורה שתוצג תהיה:

```
FOR SUM = 10 INTEREST = .2 3 YEARS
```

שורה 60 — המחשב יציג על המסך את הטקסט המופיע בגרשיים ואת ערך המשתנה המספרי P.
השורה שתוצג תהיה:

```
FUTURE VALUE IS 17.28
```

סיום התכנית — עם ביצוע הוראה 60, המחשב לא ייתקל בהוראות נוספות לביצוע, ועל־כן יסיים את ביצוע התכנית.

לפיכך ביצוע התכנית יגרום להצגת שתי השורות הבאות על המסך:

```
FOR SUM =  10 INTEREST = .2 3 YEARS
FUTURE VALUE IS 17.28
```

LIST — הוראת הצגת התכנית הנמצאת בזיכרון.
הוראה זו (בדומה להוראה RUN) איננה חלק מהתכנית, אלא הוראה חיצונית הגורמת להצגת התכנית הנמצאת בזיכרון על גבי הצג.
מבנה ההוראה:

```
LIST
LIST <LN>
LIST <LN1> — <LN2>
```

המקרה הראשון יגרום להצגת התכנית כולה על גבי המסך (בתנאי שאורך התכנית קצר ממספר השורות במסך).
המקרה השני LIST<LN> יגרום להצגת שורה מספר LN על גבי המסך.
המקרה השלישי יגרום להצגת קטעים מהתכנית, כדלקמן:

— LIST <LN1> יגרום להצגת קטע התכנית משורה שמספרה LN1, ומכאן עד סוף התכנית.
LIST — <LN2> יגרום להצגת קטע התכנית מתחילתה ועד לשורה LN2.
LIST <LN1> — <LN2> יגרום להצגת קטע התכנית משורה LN1 ועד שורה LN2.

הוספת שורה לתכנית

קיימת האפשרות להוסיף שורה חדשה לתכנית קיימת.
הוספת שורה חדשה בין שתי שורות קיימות, נעשית על־ידי מספור השורה החדשה במספר הגדול ממספר השורה שלפניה והקטן ממספר השורה שאחריה. לפני ביצוע התכנית, מסדר המחשב את השורות בסדר עולה, ועל־כן השורה החדשה תמוקם במקום המתאים.
למשל, הוספת שורה בין השורות הקיימות 30 ו־40 תיעשה על ידי מספור השורה החדשה במספר שורה 35 וכתיבת ההוראה החדשה. לפני ביצוע התכנית, המחשב יסדר את השורות בסדר עולה ועל כן שורה 35 תבוא למקומה אחרי השורה 30 ולפני השורה 40.
שים לב שהוספת שורות לתכנית אפשרית רק אם נעשה שימוש בנוהג הקיים של מספור השורות בדילוגים, כך שבין כל שתי שורות נשארים מספרים פנויים המאפשרים הוספת שורות.
הקשת השורה החדשה לא חייבת להיעשות במקומה של השורה, אלא בכל מקום, מאחר שהמחשב מסדר בעצמו את השורות בסדר עולה.

ביטול שורה מתכנית

ביטול שורה קיימת בתכנית נעשה על־ידי הקשת מספר השורה ולחיצה על מקש RETURN מיד לאחר מכן. אם קיימת בתכנית שורה בעלת מספר זהה — תיעלם שורה זו מהתכנית.

החלפת שורה בתכנית

שורה קיימת בתכנית (הכוללת הוראה כלשהי) ניתנת להחלפה על ידי כתיבת מספר השורה הקיימת לצד ההוראה החדשה האמורה להחליף את ההוראה הקיימת. בעת סידור התכנית לפי סדר מספרי השורות, תחליף השורה החדשה (שמספרה זהה למספר שורה קיימת בתכנית) את השורה הקיימת ותבוא במקומה. למעשה, הכוונה לבצע הוראה אחרת במקום ההוראה הקיימת באותה שורה.

REM — (קיצור של remark) — שורת הערה.

ההערה מאפשרת לתכניתן לשלב הסברים והערות בתוך התכנית, ובכך להקל על הבנתה. במלים אחרות, ה־REM מאפשרת "תיעוד" בגוף התכנית.
התכנית מתייחסת לשורה כזו כאל הערה שאיננה הוראת ביצוע. השורה מוצגת עם התכנית, אך אין המחשב מתייחס לשורה זו בזמן ביצוע התכנית, וממשיך בביצוע ההוראה העוקבת.
מבנה ההוראה:

<LN> REM ANY TEXT

דוגמה:

5 REM PROGRAM TO COMPUTE FUTURE VALUE

END — הוראת סיום התכנית.

ההוראה END מסיימת את ביצוע התכנית במחשב. ההוראה יכולה להופיע בכל מקום בתכנית שבו יש להפסיק את הביצוע.
מבנה ההוראה:

<LN> END

הערה:

אם התכנית צריכה לסיים ביצוע אחרי ביצוע ההוראה האחרונה בתכנית (כלומר — אחרי ביצוע ההוראה בעלת מספר השורה הגבוה ביותר) — יהיה השימוש בהוראת END מיותר, כיוון שהמחשב יפסיק לבצע את התכנית משלא ייתקל בהוראות נוספות; למען הסדר הטוב, רצוי להוסיף את ההוראה END כהוראה אחרונה בתכנית.

READ

הוראת הקלט הגורמת לקריאת נתונים מתוך רשימת נתונים. הרשימה מוגדרת על ידי ההכרזה DATA. לאחר מלת המפתח READ, תבוא רשימה של משתנים המופרדים בתווי פסיק (,) (אסור לשים פסיק בראש הרשימה או בסופה!). קריאת הערכים המועברים למשתנים נעשית מתוך רשימת נתונים, המוגדרת על־ידי הוראות DATA (ראה להלן); המשתנים מקבלים את הערכים הנקראים מרשימת הנתונים לפי סדר הופעתם.

מבנה ההוראה:

<LN> READ <V1>, <V2>, ...

דוגמה:

```
150 READ X,Y,Z, X1,Y2,Q9
```

DATA

אינה הוראת ביצוע אלא הכרזה (declaration). שורות ה־DATA מופיעות בתכנית, אך אינן משתתפות בביצועה. הכרזה זו מכילה רשימת נתונים לשימוש הוראת הקלט READ שבתכנית. בהמשך השורה מופיעה רשימת קבועים המופרדים בפסיקים. קובץ הנתונים כולו נוצר מכל שורות ה-DATA, לפי סדר עולה של מספריהן.

מבנה השורה:

<LN> DATA <C1>, <C2>,...

דוגמה:

```
330 DATA 4,2,1,7,6.73E—3,—174.1
```

בשעת ביצוע כל הוראת קריאה (READ), מתקדם בתוך קובץ הנתונים מחוון (pointer) המורה תמיד על הנתון הבא המועמד לקריאה. אם נעשה ניסיון לקרוא לאחר שהמחוון הגיע כבר לסוף קובץ הנתונים, תודפס על־ידי התכנית הודעת שגיאה:

OUT OF DATA

או:

NOT ENOUGH DATA

לאחר הדפסת הודעת השגיאה יופסק ביצוע התכנית.

RESTORE

כשיש צורך לחזור ולקרוא את כל הקובץ מתחילתו, ניתן לשבץ את ההוראה RESTORE בתכנית. ביצוע הוראה זו מחזיר את המחוון לראש הקובץ.

דוגמה:

```
340 RESTORE
```

תרגיל דוגמה 2.
נתייחס שוב לדוגמה הקודמת — תכנית לחישוב הערך העתידי.
אם התכנית הנ"ל עדיין נמצאת בזיכרון המחשב — הקש את ההוראה LIST, והתכנית תוצג לפניך.
אם התכנית הנ"ל איננה נמצאת בזיכרון — הקש אותה שוב, בצע אותה באמצעות ההוראה RUN, ולאחר מכן, הקש את ההוראה LIST, והתכנית תוצג לפניך.

א. הוספת כותרת לתכנית — שורת הערה.
הקש את השורה הזו:

```
5 REM PROGRAM TO COMPUTE FUTURE VALUE
```

בגמר הקשת השורה, הקש את ההוראה LIST. התכנית תוצג לפניך, כאשר השורה הראשונה שבה היא שורה 5.

ב. החלפת הפקודות LET בפקודה READ:
הקש את השורה הזו:

```
10 READ C,R,N
```

הקש את מספרי השורות 20 ו־30 (אחרי כל מספר שורה — זכור ללחוץ על מקש ה-RETURN.
פעולה זו תגרום לביטול שורות 20 ו־30.

ג. הוספת "מאגר הנתונים לקריאה" — ההכרזה DATA:
הקש את השורה הזו:

```
70 DATA 10,0.2,3
```

ד. הוספת ההוראה END (שאיננה הכרחית מאחר שהיא ההוראה האחרונה בתכנית).
הקש:

```
80 END
```

בסיום הקשת השורות דלעיל, הקש את ההוראה LIST, והתכנית החדשה תוצג על גבי המסך.

```
 5 REM PROGRAM TO COMPUTE FUTURE VALUE
10 READ C,R,N
40 LET P = C*(1+R)**N
50 PRINT "FOR SUM =" ;C, "INTEREST =" ;R,N ; "YEARS"
60 PRINT "FUTURE  VALUE IS" ; P
70 DATA 10,0.2,3
80 END
```

הפעלת תכנית זו על ידי הקשת RUN תיתן תוצאה הזהה לתוצאה שבתכנית המקורית (דוגמה 1).

INPUT

הוראה זו מאפשרת להזין נתונים תוך מהלך ביצוע התכנית. דבר זה דרוש כאשר הנתונים מוזנים לתכנית דרך המסוף.

מבנה ההוראה: לאחר מלת המפתח INPUT מופיעה רשימת משתנים המופרדים בתווי פסיק:

<LN> INPUT <V1>, <V2>, <V3> ...

בהגיע התכנית להוראה זו, היא מציגה על המסוף סימן שאלה (?), פעולתה נעצרת והיא ממתינה לקבלת הנתונים. על מפעיל המסוף להזין את הנתונים על־ידי הקשת רשימה של קבועים במספר המתאים למספר המשתנים המופיעים לאחר המלה INPUT. ערכי הקבועים שהמשתמש מקיש חייבים להיות מופרדים בתווי פסיק. לאחר הקשת הנתונים על המשתמש ללחוץ על מקש המשלוח שעל גבי המסוף (RETURN) הגורם להעברת הנתונים לתכנית, ואז תציב התכנית את ערכי הקבועים למשתנים המתאימים ותמשיך בעיבוד. אם המשתמש מקיש מספר ערכים גדול מדי, יודיע המחשב על כך, והמשתמש יתבקש להזין את הנתונים שוב. אם מספרם קטן מדי, יודיע המחשב על כך והמשתמש יתבקש להוסיף את הנתונים החסרים.

רצוי שאת ההוראה INPUT תקדם ההוראה PRINT, אשר תציג למשתמש הנחיות לגבי הנתונים שהתכנית דורשת.

לדוגמה, אם התכנית תכלול את ההוראות האלה:

```
120 PRINT "YOUR VALUES OF X,Y AND Z ARE"
130 INPUT  X,Y,Z
```

המחשב יציג:

```
YOUR VALUES OF X,Y, AND Z ARE
?
```

לאחר מכן ימתין המחשב להכנסת שלושת הנתונים על־ידי המשתמש. סימן השאלה הוא כתוצאה מההוראה INPUT. ההוראה שהוצגה למשתמש לפני הצגת סימן השאלה מבהירה לו שעליו להזין שלושה נתונים עבור משתנים X, Y ו־Z. אפשרות אחרת היא לכלול את הוראות ההקשה בתוך הוראת ה־INPUT, במבנה הזה:

דוגמה:

```
120 INPUT "TYPE X,Y,Z" ; X,Y,Z
```

עם ביצוע הוראה וו בתכנית יוצג על המסך —

```
TYPE X ,Y, Z
```

והמחשב ימתין לקבלת שלושה ערכים מספריים. משיקלטם ישייך אותם בהתאמה למשתנים המספריים X, Y, Z.

תרגיל דוגמה 3

נתייחס לתכנית שבדוגמה 2 ונחליף את ההוראה READ — הקוראת נתונים ממאגר נתונים פנימי (הכרזות DATA) — בהוראה INPUT, שתאפשר למשתמש להקיש ערכים מספריים חיצוניים לחישוב הערך העתידי.

א. החלפת ההוראה READ בהוראה INPUT:

הקש את השורה הזו:

```
10 INPUT "TYPE SUM, INTEREST, YEARS" ; C,R,N
```

ב. בטל את שורת ה־DATA:

הקש את מספר השורה 70 (ולאחריו לחץ על מקש ה־RETURN)

הקש את ההוראה LIST, והתכנית תוצג על גבי המסך:

```
 5 REM PROGRAM TO COMPUTE FUTURE VALUE
10 INPUT "TYPE SUM, INTEREST, YEARS" ; C,R,N
40 LET P=C* (1+R)**N
50 PRINT "FOR SUM =" ; C, "INTEREST =" ; R,N ; "YEARS"
60 PRINT "FUTURE VALUE IS" ;P
80 END
```

הפעלת תכנית זו על ידי הקשת ההוראה RUN תציג על המסך את ההודעה הזו:

```
TYPE SUM, INTEREST, YEARS
```

והתכנית תצפה לקבלת שלושה נתונים: הסכום שיש לחשב את ערכו העתידי, הריבית שלפיה יעשה החישוב ומספר השנים לחישוב.

הקש את הנתונים באופן הזה:

```
TYPE SUM, INTEREST, YEARS 10,0.2,3
```

ולחץ על מקש RETURN. התוצאה תהיה זהה לתוצאה שבתכניות הקודמות. הפעל תכנית זו שוב (על־ידי ההוראה RUN) והקש ערכים חדשים לסכום, לריבית ולמספר השנים. התכנית תציג מיד את הערך העתידי התואם לערכים החדשים.

GOTO — הוראת הסתעפות (branch)

לעתים נדרש שהתכנית לא תתבצע לפי סדר הופעת ההוראות, אלא תסתעף להוראה הנמצאת במקום אחר בתכנית. הסתעפות זו ניתנת לביצוע על־ידי שימוש בהוראה GOTO.

צורת ההוראה:

```
<LN> GOTO <LN1>
```

משמעות הוראה זו העברת המשך ביצוע התכנית להוראה בשורה מספר <LN1>.

דוגמה:

```
150 GOTO 75
```

תרגיל דוגמה 4

נמשיך להתייחס לדוגמה הקודמת.

א. הוסף הוראת GOTO אשר תחזיר את ביצוע התכנית להתחלה לאחר ביצוע החישוב והצגתו.

הקש את השורה הבאה:

```
70 GOTO 10
```

בעת הביצוע, תחשב התכנית את הערך העתידי עבור הסכום, הריבית ומספר השנים שתקיש כמענה לשאלת התכנית, ותשוב ותבקש ערכים חדשים לביצוע חישוב נוסף.

שים לב: תכנית זו לא תסיים את הביצוע — היא תמשיך ותבקש ערכים חדשים! להפסקת הביצוע כבה את המסוף.

IF THEN — הוראת התנאי

הוראה זו מאפשרת להסתעף מהסדר הרגיל של ביצוע הוראות התכנית, כאשר ביטוי יחס מסוים מתקיים, אחרת — יימשך ביצוע התכנית בצורה הרגילה, כלומר תבוצע ההוראה הבאה.

צורתה הכללית של הוראה זו תהיה:

```
<LN> IF <REL> THEN <LN1>
```

<REL> הוא ביטוי יחס המשמש כתנאי.

<LN1> הוא מספר שורה של הוראה כלשהי בתכנית (ההוראה <LN1> מכונה "ההוראה המותנית")

משמעות ההוראה היא שהמשך ביצוע התכנית יעבור לשורה <LN1> אם התנאי <REL> הוא אמתי. ולא, תתבצע ההוראה שבשורה העוקבת.

דוגמאות:

```
340 IF X<=M THEN 630
420 IF G=0 THEN 165
```

בשורה 340, אם ערך המשתנה המספרי X קטן או שווה לערך המשתנה המספרי M, תסתעף התכנית להוראה 630, ולא, תתבצע ההוראה הבאה בתכנית.

בשורה 420, אם המשתנה המספרי G שווה לאפס, תסתעף התכנית להוראה 165, ולא, תתבצע ההוראה הבאה בתכנית.

בעזרת הוראה זו ניתן להסתעף לכל שורה בתכנית — הנמצאת לפני או לאחרי השורה שבה נבדק התנאי.

שימוש אחר בהוראת התנאי נעשה לביצוע פעולה מותנה.

מבנה ההוראה:

<LN> IF <REL> THEN <EXP>

<LN> IF <REL> THEN <STATEMENT>

<EXP> הוא ביטוי אריתמטי

<STATEMENT> הוא הוראה ב־BASIC.

דוגמאות:

```
280 IF X<Y THEN Z=X+Y
320 IF G=0 THEN G=6
680 IF A=5 THEN PRINT "A=5"
```

בשורה 280, אם ערך המשתנה X קטן מערך המשתנה המספרי Y, תחשב התכנית את הסכום X+Y, תעביר את התוצאה למשתנה המספרי Z, ותמשיך להוראה הבאה בתכנית. ולא, (X גדול או שווה ל־Y) לא תשנה התכנית את הערך של Z, ותעבור להוראה הבאה בתכנית.

בשורה 320, אם הערך של G הוא 0, תעביר התכנית את הערך 6 ל־G, ותמשיך להוראה הבאה. אם הערך של G איננו 0, לא תשנה התכנית את ערכו של G ותעבור להוראה הבאה.

בשורה 680 אם הערך של A הוא 5, תציג התכנית את ההודעה A=5.

תרגיל — דוגמה 5

נתייחס לדוגמה 4.

א. הוסף לתכנית הוראת הסתעפות, כך שהתכנית תסיים ביצוע, אם ערך הסכום הנוכחי שיוקש לבקשת התכנית יהיה 0.
הקש את השורה הזו:

```
20 IF C=0 THEN 80
```

התכנית תתבצע כמספר הפעמים שתדרוש ותסיים את הביצוע כאשר הסכום שיוקש כנתון יהיה 0. שים לב: גם לסיום התכנית יש להקיש שלושה נתונים — הסכום, הריבית ומספר השנים.

התכנית נראית כך:

```
 5 REM PROGRAM TO COMPUTE FUTURE VALUE
10 INPUT "TYPE SUM, INTEREST, YEARS" ; C.R.N
20 IF C=0 THEN 80
40 LET P=C*(1+R)**N
```

```
50 PRINT "FOR SUM=" ;C, "INTEREST =" ;R,N; "YEARS"
60 PRINT "FUTURE VALUE IS" ; P
70 GOTO 10
80 END
```

ON-GOTO — הוראת ההסתעפות הרב־כיוונית.

מבנה ההוראה:

<LN> ON <EXP> GOTO <LN1>, <LN2>,... <LNm>

LNi הם מספרי שורות (פסיק ביניהם).
כאשר מגיע הביצוע להוראה זו, מחושב הערך של EXP ונקטע למספר שלם. מספר שלם זה חייב להיות בין 1 לבין מספר מספרי השורות המופיעים לאחר המלה GOTO (כלומר $m \geqslant$ ערך EXP $\geqslant 1$)

אם הערך השלם קטן מ־1 או גדול מ־m תופיע הוראת שגיאה כדלקמן:
ON STATEMENT OUT OF RANGE

ההסתעפות תתבצע לשורה LNi כאשר הערך השלם של <EXP> הוא i.

דוגמה:

```
230 ON X+Y GOTO 575, 490, 150
```

הוראה זו תחשב את ערך הביטוי X+Y, ובהתאם לתוצאה תקבע את מספר ההוראה שאליה תסתעף התכנית.
אם התוצאה אינה מספר שלם, תקטע התכנית את החלק העשרוני של התוצאה ותתייחס רק לחלק השלם. למשל, אם התוצאה היא 2.7, התכנית תתייחס אליה כאילו היא 2.
אם ערך הביטוי 1, תסתעף התכנית להוראה 575.
אם ערך הביטוי 2, תסתעף התכנית להוראה 490.
אם ערך הביטוי 3, תסתעף התכנית להוראה 150.
אם ערך הביטוי קטן מ־1 או גדול מ־3 תדפיס התכנית הודעת שגיאה.

FOR-TO-STEP — הוראה להתחלת לולאה

NEXT — הוראה לסיום לולאה.
לעתים נדרש לבצע קטע כלשהו של התכנית מספר רב של פעמים. הוראת הלולאה (LOOP) מורכבת משתי הוראות נפרדות, שביניהן סדרת ההוראות המבוקשת. היא מאפשרת את ביצועה של סדרת הוראות במספר הפעמים הנדרש. הלולאה מתחילה בהוראת התחלת הלולאה, התכנית מבצעת את ההוראות הנמצאות אחרי הוראה זו, וכאשר מגיעה התכנית להוראת סיום הלולאה, חוזרת התכנית ומבצעת

שוב את ההוראות הנמצאות אחרי הוראת התחלת הלולאה, עד שהוראות אלה מבוצעות כמספר הפעמים הנדרש. הצורה הכללית של ההוראה היא:

<LN> FOR <V> =<EXP1> TO <EXP2> STEP <EXP3>

• סדרת
•
•
• הוראות

<LN> NEXT <V>

לכל הוראה של התחלת לולאה המופיעה בתכנית חייבת להיות גם הוראת סיום לולאה. אם חסרה אחת מהן, תודיע התכנית על שגיאה. הוראת סיום לולאה חייבת להופיע אחרי הוראת התחלת הלולאה.
<V> חייב להיות משתנה מספרי (לא ביטוי!); הוא נקרא "משתנה הלולאה".
<EXP1>, <EXP2> <EXP3> יכולים להיות ביטויים אריתמטיים כלשהם.
במהלך הביצוע, כאשר התכנית מגיעה להוראה של התחלת לולאה, מחושבים הערכים של שלושת הביטויים האריתמטיים הנ"ל לפי ערכם המספרי של הקבועים והמשתנים באותו זמן.
הערך של <EXP1> נקרא "ערך התחלה" של <V>. הערך של <EXP2> נקרא "הערך הסופי" שלו, וערכו של <EXP3> הוא "ערך הצעד" (step value) של <V>. ערך זה יכול להיות מספר חיובי או שלילי (אך לא אפס!).

הערה: אם STEP <EXP3> לא מופיע בהוראת התחלת הלולאה, יהיה ערך הצעד 1 +.

שלושת הערכים הללו אינם משתנים בזמן ביצוע הלולאה. הם יכולים להשתנות רק אם נפנה חזרה אל הוראת ההתחלה. אפילו אם נשנה בתוך הלולאה את הערכים של המשתנים המרכיבים את <EXP1>, <EXP2> ו־<EXP3>, תעבוד הלולאה לפי שלושת הערכים שחושבו ברגע ההתחלה. (דבר זה איננו נכון לגבי משתנה הלולאה, אשר אפשר לשנות את ערכו באמצעות ההוראות שבתוך הלולאה).

ערך משתנה הלולאה <V> "נבדק" בסיום כל מחזור של הלולאה. פירוש הדבר, שכאשר מגיעים לסוף הלולאה מוסיפה התכנית לערך משתנה הלולאה את ערך הצעד ובודקת אם כתוצאה מכך "עבר" ערך המשתנה את הערך הסופי. אנו משתמשים כאן במלה "עבר" משום שהצעד יכול להיות חיובי או שלילי. אם ערך הצעד חיובי, פירוש הדבר כי ערך המשתנה גדל מעל הערך הסופי (אם הוא שווה — לא עבר). אך אם ערך הצעד שלילי, פירושו של "עבר" הוא כי ערך המשתנה קטן מהערך הסופי. כלומר, כאשר מסתיים ביצוע הלולאה, יהיה למשתנה הלולאה הערך האחרון שעדיין אינו עובר את הערך הסופי. בסיום ביצוע הלולאה ממשיכה התכנית להוראה הבאה אחרי הוראת סיום הלולאה.

דוגמאות:

א.

```
456 FOR J=1 TO 12
●
●
●
480 NEXT J
```

הערך ההתחלתי של משתנה הלולאה J הוא 1, גודל הצעד הוא 1, הלולאה תתבצע 12 פעמים והערך הסופי של משתנה הלולאה יהיה 12.

ב.

```
230 FOR F3 = —3 TO 12 STEP 2
●
●
●
270 NEXT F3
```

הערך ההתחלתי של משתנה הלולאה F3 הוא 3—, גודל הצעד הוא 2, הלולאה תתבצע 8 פעמים והערך הסופי של משתנה הלולאה יהיה 11.

ג.

```
100 FOR X=1 TO 10 STEP 2
110 LET X = X*2
●
●
●
150 NEXT X
```

הערך ההתחלתי של משתנה הלולאה X הוא 1, גודל הצעד הוא 2, אך משתנה הלולאה משנה את ערכו גם בתוך הלולאה, ולכן הלולאה תתבצע רק 3 פעמים והערך הסופי של משתנה הלולאה יהיה 9.

ד.

```
130 FOR A=0 TO 3 STEP D
●
●
●
180 NEXT A
```

הערך ההתחלתי של משתנה הלולאה A הוא 0, גודל הצעד כערך המשתנה המספרי D (שחייב להיות גדול מ־0). מספר הפעמים שהלולאה תתבצע תלוי בערך של D, וכך גם ערכו הסופי של משתנה הלולאה.

ה.

```
340 FOR K2 =(17+Z*2)/3 TO 3*Z STEP 1/4
•
•
•
380 NEXT K2
```

גודל הצעד הוא 0.25, הערך ההתחלתי של משתנה הלולאה, מספר הפעמים שהלולאה תתבצע, והערך הסופי של משתנה הלולאה תלויים בערך המשתנה המספרי Z.

ו.

```
150 FOR S =2 TO —10
•
•
•
170 NEXT S
```

הערך ההתחלתי של משתנה הלולאה הוא 2, גודל הצעד הוא 1, הלולאה תתבצע פעם אחת בלבד (כיוון שהערך הסופי קטן מהערך ההתחלתי כבר בפעם הראשונה), והערך הסופי של משתנה הלולאה יהיה 2.

חוקי השימוש בהוראות הלולאה

1. מותר לשנות בתוך הלולאה את ערכו של משתנה הלולאה, אך יש לשים לב שהדבר עשוי לשנות את מספר הפעמים שהלולאה מתבצעת (ראה דוגמה ג׳).

2. מותר לצאת מלולאה, לבצע הוראות שאינן כתובות בתוך הלולאה ולחזור אליה, בתנאי שהדבר הגיוני — כלומר שאפשר להמשיך ולבצע את הלולאה שאליה חוזרים לפי הערך הסופי וערך הצעד שנקבעו בכניסה הראשונה ללולאה.

מובן שמותר לצאת מלולאה וכלל לא לחזור אליה. במקרה כזה ערך משתנה הלולאה יהיה ערכו ברגע היציאה מהלולאה.

3. בתוך לולאה אחת יכולה להופיע לולאה שנייה, בתנאי שהלולאה הפנימית כלולה כולה בתוך הלולאה החיצונית. במקרה כזה, בכל מעבר של הלולאה החיצונית, תתבצע הלולאה הפנימית במלוא מספר מעבריה. מספר הלולאות הפנימיות היכולות להופיע בתוך לולאה חיצונית הוא לכל היותר 10.

דוגמא:

```
30 FOR I = 1 TO 10
•
•
50 FOR J = —5 TO 30 STEP 2
•
•
90 FOR I3 = 1 TO 10+I
•
•
110 NEXT I3
•
•
•
150 NEXT J
•
•
190 FOR J = 1 TO I
•
•
230 FOR K = 100 TO I STEP —2
•
270 NEXT K
•
•
310 FOR K = I TO 100 STEP I
•
•
340 NEXT K
•
400 NEXT J
•
480 NEXT I
```

יש לשים לב שההתחלה והסיום של כל לולאה פנימית כלואים שניהם בתוך הלולאה החיצונית. אם תנאי זה אינו מתקיים תיווצר הודעת שגיאה בנוסח:

FOR WITHOUT NEXT

או

NEXT WITHOUT FOR

לצורת תכנית זו קוראים loop nesting.

4. אם צריך לפסוח (בתנאים מסוימים) על ביצוע חלק מההוראות בלולאה, יש להקפיד להפנות תמיד את הביצוע אל הוראת הסיום של אותה לולאה. אם יופנה הביצוע לראש הלולאה, תראה בכך התכנית כוונה להתחיל מחדש בביצוע הלולאה.

למשל, בקטע התכנית המובא להלן יודפסו המספרים מ־1 עד 10 פרט למספר 6.

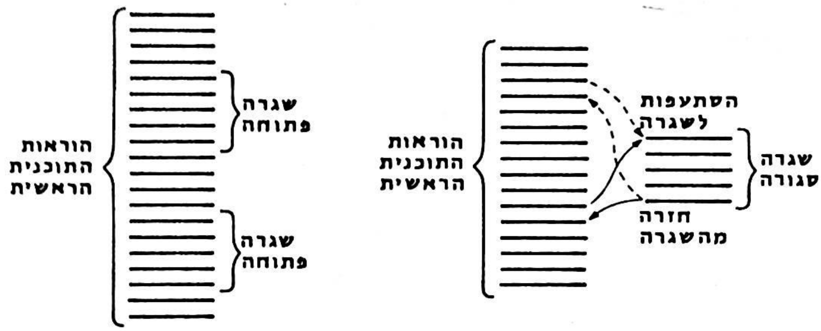
```
10 FOR I = 1 TO 10
20 IF I = 6 THEN 40
30 PRINT I
40 NEXT I
```

GOSUB הוראת הסתעפות לשגרה סגורה.

RETURN הוראת חזרה משגרה סגורה.
לעתים קיימת סדרת הוראות בתכנית, המבצעת תהליך מסוים ומופיעה באותה צורה במקומות שונים בתכנית. לדוגמה — קטע תכנית המקבל סדרת ערכים מספריים ומבצע חישוב הממוצע של ערכים אלה, יכול להופיע בכמה מקומות בתכנית, ובכל מקום לבצע את חישוב הממוצע לסדרת מספרים שונה. בכל מקום שבו יופיע קטע זה, תופיע למעשה אותה סדרת הוראות. קטע זה נקרא שגרה (subroutine).

כאשר קטע כזה כתוב בכמה מקומות בתכנית הוא נקרא שגרה פתוחה. קיימת אפשרות לחסוך בכתיבת הקטע מספר פעמים על־ידי כתיבתו פעם אחת בלבד בתכנית. משנדרש לבצע את קטע ההוראות, תסתעף התכנית למקום שבו נמצא הקטע הדרוש, תבצע את ההוראות המופיעות בו, ותחזור להמשך ביצוע התכנית מהמקום שאליו הסתעפה אל אותו קטע. קטע כזה, הכתוב פעם אחת בלבד

בתכנית ושהתכנית מסתעפת לשם ביצועו, נקרא שגרה סגורה (תרשים 3.1)



תרשים 3.1 שגרה פתוחה ושגרה סגורה

ההוראה GOSUB המופיעה בתכנית הראשית "קוראת" לשגרה הסגורה. הוראות השגרה מתבצעות, ובסיום קטע ההוראות ממשיכה התכנית לבצע את ההוראה המופיעה אחרי ה-GOSUB.
ההסתעפות לשגרה סגורה נעשית על־ידי ציון מספר השורה שבה מתחילה השגרה.

צורת ההוראה היא:

<LN> GOSUB <LN1>

דוגמה:

85 GOSUB 1000

ההוראה RETURN חייבת להיות ההוראה האחרונה בשגרה הסגורה. כאשר התכנית תבצע את ההוראות המופיעות בשגרה ותגיע להוראה זו, תחזור התכנית מהשגרה ותבצע את ההוראה העוקבת אחרי הוראת ה־GOSUB אשר גרמה להסתעפות לשגרה.
צורת ההוראה היא:

<LN> RETURN

דוגמה:

1200 RETURN

כאמור לעיל, ניתן להסתעף לשגרה סגורה מכל מקום בתכנית, ובסיום ביצוע השגרה ימשיך הביצוע בהוראה המופיעה אחרי ההוראה שקראה לשגרה.
ניתן להסתעף לשגרה סגורה אחת מתוך שגרה סגורה שנייה ואפילו להסתעף לשגרה הסגורה עצמה, כאשר הביצוע נמצא כבר בתוכה. יש להקפיד שהתכנית לא תסתעף לתוך שורות של שגרה סגורה מבלי שנשלחה לשם על־ידי הוראת GOSUB (הדבר יגרום לשגיאה בביצוע והמחשב יודיע את ההודעה הבאה: RETURN WITHOUT GOSUB).

אין להסתעף על־ידי הוראת GOSUB לקטע תכנית שאינו שגרה סגורה, זאת אומרת לקטע שאיננו מסתיים בהוראה RETURN (השגיאה, במקרה כזה, אינה מתגלה על־ידי הודעת שגיאה, אלא על־ידי ביצוע לא תקין של התכנית!).

פונקציות

DEF FN הוראת הגדרת הפונקציה.

FN שם הפונקציה.

אם בתכנית נעשה שימוש בנוסחאות חישוב ארוכות, ואלו מופיעות באותה צורה במקומות רבים בתכנית, ומבצעות אותו חישוב על ערכים שונים בכל מקום שבו הן מופיעות, ניתן להגדיר נוסחאות אלה פעם אחת בלבד ולחסוך את כתיבתן בכל המקומות שבהם נעשה שימוש בהן. נוסחה המוגדרת פעם אחת בלבד, אבל נעשה בה שימוש במקומות שונים בתכנית, נקראת פונקציה (function).

אם, למשל, יש לחשב במקומות שונים בתכנית את הערך של הנוסחה

$$A^2X + (2X^2AB^3(3+2AX^6-B^2A/5)-10)\ X^3A^2B$$

נוח לכתוב את הנוסחה פעם אחת בלבד בתכנית, ולאחר מכן להשתמש בה באמצעות סימן מיוחד המזהה אותה. כתיבת הנוסחה פעם אחת בלבד עדיפה גם מהסיבה שהסיכוי לשגיאה בכתיבה קטן. בנוסף, אם נוסחה זו היא רק חלק מנוסחה ארוכה ומסובכת, קל בהרבה להשתמש בסימון המיוחד לנוסחה זו במקום לכתוב את כולה. הגדרת הפונקציה וקביעת שמה נעשות על־ידי ההוראה DEF FN שצורתה:

<LN> DEF FN □ (V) = <EXP>

מלת המפתח DEF פירושה "הגדר" (define) □ FN מסמל את שם הפונקציה המוגדרת: הסימן □ מייצג אות מאותיות האלפבית (האנגלי), V מייצג משתנה מספרי שהוא המשתנה של הפונקציה (ארגומנט). הנוסחה מופיעה כביטוי אריתמטי מימין לסימן השוויון. הגדרת הפונקציה מוגבלת לשורה אחת בלבד, ואין אפשרות להמשיך את ההגדרה בשורות אחרות.

דוגמאות:

```
100 DEF FNA(X) = X**2+2*X+5
150 DEF FND(A)=A*R*P +R**2*A/ (3*P + A)
```

בדוגמה הראשונה, X הוא משתנה הפונקציה; כאשר יעשה שימוש בפונקציה יוצב הערך של משתנה הפונקציה ויבוצע חישוב הנוסחה. תוצאת החישוב תוצב ל־FNA(X).

בדוגמה השנייה משתנה הפונקציה הוא A, ומשתתפים בביטוי שני משתנים מספריים אחרים — R ו־P. את ערכם של המשתנים A, R ו־P יש לקבוע בתכנית לפני השימוש בפונקציה, ואילו ערך משתנה הפונקציה נקבע בזמן השימוש בה. תוצאת החישוב תוצב ל־FND(A).

השימוש בפונקציה נעשה על־ידי כתיבת שם הפונקציה וציון משתנה הפונקציה. שם הפונקציה יכול להופיע כמשתנה מספרי בתוך ביטוי אריתמטי.
דוגמאות:

```
180 LET Y = 3*FNA(X) + 5/FND(A)
310 LET B = FNA(Z)**2
```

בדוגמה הראשונה מוצב ערך משתנה הפונקציה X ומחושב הערך של FNA (כפי שהוגדר בהוראת DEF FNA(X)), לאחר מכן מתבצע הכפל של FNA(X) ב־3, ואחר מוצב ערך משתנה הפונקציה A ומחושב הערך של FND, מתבצעת החלוקה של 5 בערך של FND(A), ולבסוף מתבצע חיבור תוצאות הביניים שהתקבלו וערך התוצאה הסופית מוצב למשתנה המספרי Y.
בדוגמה השנייה מוצב ערך משתנה הפונקציה Z ומחושב הערך של FNA, לאחר מכן מחושב הערך של FNA בריבוע, ותוצאת החישוב תוצב במשתנה המספרי B.
שים לב, שבהגדרת הפונקציה FNA משתנה הפונקציה סומן ב־X, ואילו בזמן השימוש בפונקציה משתנה הפונקציה הוא Z: התכנית מציבה לצורך החישוב את ערך משתנה הפונקציה בביטוי, ולכן מותר להשתמש בכל משתנה מספרי.
כאשר נעשה שימוש בפונקציה כבמשתנה מספרי בתוך ביטוי אריתמטי, עדיפות הפונקציה הנה כעדיפות הסוגריים, כלומר — חישוב ערך הפונקציה יתבצע לפני ביצוע האופרטורים האחרים.
בזמן השימוש בפונקציה, יכול משתנה הפונקציה להיות ביטוי אריתמטי כלשהו, והוא יכול לכלול גם פונקציות אחרות (אך לא את הפונקציה עצמה).
דוגמאות:

```
290 LET S = T + 2* FNA (T+U/2)+1/C FND(U)*T*3)
940 LET K = FND((Q*4*FNA (Q**3-M/N + Q))-2*Q)
```

קיימת אפשרות לשנות הגדרת פונקציה בזמן ביצוע התכנית על־ידי כתיבת ההוראה DEF FN לפונקציה שכבר הוגדרה. כאשר יעשה שימוש בפונקציה, הגדרתה תהיה בהתאם להגדרה האחרונה שעליה עברה התכנית.

פונקציות פנימיות

קיימות מספר נוסחאות אשר השימוש בהן נפוץ. כמה מנוסחאות אלו הוגדרו בשפת ה־BASIC, כך שהתכניתן איננו צריך להגדירן כדי להשתמש בהן בתכניתו. נוסחאות אלה נקראות פונקציות פנימיות (built-in functions), ושמן מורכב משלוש אותיות. כדי להשתמש בהן יש לכתוב את שם הפונקציה ואת משתנה הפונקציה. מותר להשתמש בפונקציות אלה כבמשתנים אריתמטיים.
משתנה הפונקציה יכול להיות ביטוי אריתמטי כלשהו, והוא יכול להכיל גם פונקציות אחרות. להלן נסמן ב־X את משתנה הפונקציה.

פונקצית השורש הריבועי (SQR(X
מקבלת את ערך השורש הריבועי של משתנה הפונקציה X אם ערכו חיובי או אפס.
אם ערכו של המשתנה X שלילי, תודפס הודעת שגיאה וביצוע התכנית יופסק.
דוגמה לשימוש:

```
320 LET S2 = SQR(FNA(M))
```

התכנית תחשב את ערך הפונקציה FNA בהתאם לערכו של M; אם ערך הפונקציה FNA אינו שלילי — תחשב התכנית את השורש הריבועי של ערך הפונקציה, ותציב את הערך שהתקבל למשתנה המספרי S2; אם ערך הפונקציה FNA שלילי — תודפס הודעת שגיאה וביצוע התכנית יופסק.

פונקצית השלם (INT(X
מקבלת את ערך המספר השלם הקרוב ביותר לערך משתנה הפונקציה X, והקטן או שווה לערך X.
דוגמאות:

INT (5) יהיה 5
INT (5.8) יהיה 5
INT (—5.2) יהיה —6

דוגמאות לשימוש:

```
85 LET M2 = INT (M1 + 0.5)
```

המשתנה המספרי M2 יקבל את הערך המעוגל למספר שלם של המשתנה המספרי M1.

```
290 IF X/Y = INT (X/Y) THEN 100
```

אם ערך המשתנה המספרי Y הוא מחלק של ערך המשתנה המספרי X (כלומר X/Y, מתחלק ללא שארית), תסתעף התכנית להוראה 100. ולא, תמשיך התכנית להוראה הבאה.

פתקצית הסימן (SGN(X
מקבלת ערך 1 אם ערך משתנה הפונקציה X חיובי.
מקבלת ערך 0 אם ערך משתנה הפונקציה X אפס.
מקבלת ערך 1— אם ערך משתנה הפונקציה X שלילי.
דוגמה לשימוש:

```
120 ON SGN(A) + 2 GO TO 80, 190, 40
```

אם ערך המשתנה המספרי A שלילי, תסתעף התכנית להוראה 80; אם ערכו אפס, התכנית תסתעף להוראה 190, ואם ערכו חיובי תסתעף התכנית להוראה 40.

פונקצית הערך המוחלט ABS(X)
מקבלת את ערך המשתנה X אם ערכו חיובי או אפס.
מקבלת את הערך הנגדי (כלומר X—) אם ערכו שלילי.
לכן, ABS(X) תקבל תמיד ערך חיובי או אפס.
דוגמאות לשימוש:

```
90  LET Y5 = ABS(Y5)
```

אם ערך המשתנה Y5 חיובי או אפס — ערכו לא ישתנה.
אם ערך המשתנה Y5 שלילי — ערכו יישאר אותו מספר, אך בשינוי סימן (כלומר, הערך החדש של Y5 יהיה חיובי).

```
400 LET D7 = ABS(D1 — D2)
```

אם ערך המשתנים D1 ו־D2 מקיים D1>D2, יקבל המשתנה D7 את הערך D2 — D1. אם ערכם מקיים D1<D2, יקבל המשתנה D7 את הערך D1 — D2.
כלומר ערכו של D7 יהיה המרחק בין הערכים של D1 ו־D2.

פונקציות הערך האקראי RND(0)
מקבלת ערך מקרי בתחום הפתוח 1—0. משתנה הפונקציה הוא תמיד אפס.
דוגמאות לשימוש:

```
100 LET X = RND(0)
```

X יהיה מספר אקראי בין 0 ל־1.

```
350 LET Y = N*RND(0)
```

Y יהיה מספר אקראי בין 0 ל־N.

```
190 LET P2 = A + (B—A)*RND(0)
```

P2 יהיה מספר אקראי בין A ל־B (A<B)

```
230 LET Q = INT (N*RND (0))
```

Q יהיה מספר אקראי שלם בין 0 ל־N-1 (N מספר שלם).

```
170 Z = M + INT ((N-M)*RND(0))
```

Z יהיה מספר אקראי שלם בין M ל־N-1 (M ; N מספרים שלמים, N>M)

פונקצית הסינוס SIN(X)
מקבלת את הערך של סינוס X.
הערך של X חייב להיות ברדיאנים (כלומר $0 \leq X \leq 2\pi$)

פונקציות הקוסינוס (COS(X

מקבלת את הערך של קוסינוס X.
הערך של X חייב להיות ברדיאנים.

פונקציות הטנגנס (TAN(X

מקבלת את הערך של טנגנס X.
הערך של X חייב להיות ברדיאנים.

פונקציות הטנגנס ההפוכה (ATN(X) (arctangent

מקבלת את הערך ברדיאנים של הטנגנס ההפוך של X.
עבור X כלשהו מחזירה הפונקציה ערך בין π/2 — לבין π/2

פונקציות הלוגריתמוס הטבעי (LOG (X) LN(X

מקבלת את הערך של הלוגריתמוס הטבעי של X (על בסיס e).
הערך של X חייב להיות חיובי.

פונקצית המעריך הטבעי (EXP(X) (e

מקבלת את הערך E^X (E = 2.71828).

תרגיל — דוגמה 6

התכנית תקבל כקלט סכום כסף C, ותפיק טבלה המציגה את הערך העתידי לתקופות של שנה, שנתיים, שלוש, ארבע, וחמש שנים, ובשער ריבית 0.1, 0.2, 0.3, 0.4, 0.5.

```
 5 REM PROGRAM TO COMPUTE FUTURE VALUE
 7 DEF FNX(Y)= INT (0.5 + 1000*Y)/1000
10 INPUT "TYPE SUM" ; C
15 IF C = 0 THEN 100
20 PRINT "FUTURE VALUE OF SUM" ; C
25 PRINT "R=",
30 FOR R = 0.1 TO 0.5 STEP 0.1
35 PRINT R,
40 NEXT R
45 PRINT
50 FOR N = 1 TO 5
55 PRINT "N=" ; N,
60 FOR R = 0.1 TO 0.5 STEP 0.1
```

```
 65 PRINT FNX (C*(1+R)**N),
 70 NEXT R
 75 PRINT
 80 NEXT N
 85 PRINT
 90 GO TO 10
100 END
```

מחרוזות

מחרוזת (string)

בנוסף לטיפול במספרים, בקבועים מספריים, במשתנים מספריים ובביטויים אריתמטיים מאפשרת שפת ה־BASIC לטפל גם במחרוזות. מחרוזת היא סדרת תווים כלשהי המסוייגת בסימני גרשיים(״). השימוש בתכנית במחרוזות נעשה לצורך הדפסת כותרות, שמות, הסברים והודעות קבועות כלשהן. לא ניתן לבצע פעולות אריתמטיות על מחרוזות, מאחר שהן חסרות כל משמעות מספרית או כמותית.

דוגמאות של מחרוזות:

"123"
"THIS IS STRING CONSTANT"
"ABC123 DEF456"

משתנה מחרוזתי (string variable)

זהו המשתנה המקבל ״תוכן״ של מחרוזת, ויכול לשנות את ״תוכנו״ בזמן ביצוע התכנית.

סימון משתנה מחרוזתי ב־BASIC נעשה בתו אלפביתי ולאחריו (מימינו) בתו הדולר ($).

דוגמה:

B$
X$
A$

הערה: התכנית מבחינה בין המשתנה A שהוא משתנה מספרי לבין המשתנה A$ שהוא משתנה מחרוזתי.

אין להשתמש במשתנים מחרוזתיים בביצוע פעולות אריתמטיות.

היחסים בין מחרוזות

אופרטורי־היחס המשמשים לקביעת יחסים בין משתנים מספריים, משמשים גם למחרוזות.

A$ = B$ פירושו, שהמחרוזת A$ זהה למחרוזת B$.

A$<>B$ פירושו שהמחרוזות אינן זהות.

A$ < B$ פירושו שהמחרוזת A$ קודמת בסדר הלקסיקוגרפי (אלפביתי ו/או מספרי) למחרוזת B$.

A$ > B$ פירושו שהמחרוזת B$ קודמת בסדר הלקסיקוגרפי למחרוזת A$.

דוגמאות:

"ABC" > "AB"

"AZB3" > "A1B2" (כיון שהאותיות קודמות למספרים)

"ST " > "ST" (כיון שהמחרוזת הימנית ארוכה יותר — מכילה תווי רווח).

היחסים בין המחרוזות, ככל היחסים האחרים, הם ביטויים לוגיים בעלי ערך של אמת או שקר ויכולים לשמש בכל מקום שבו מותר להשתמש בהם, למשל בהוראת התנאי 'כד'.

אין להשוות משתנים מספריים עם משתנים מחרוזתיים.

הוראות הקשורות לטיפול במחרוזות

השימוש במחרוזות מותר רק בחלק מהוראות ה־BASIC. הוראות אלה הן: READ ,DATA ,PRINT ,LET ,INPUT ,IF־THEN.

להלן הכללים לשימוש בהוראות לצורך טיפול במחרוזות:

READ, DATA

הוראות אלה זהות להוראות READ ו־DATA שתוארו לעיל. בין המשתנים הבאים ברשימה המופיעה אחרי מלת המפתח READ, יכולים להופיע גם משתני מחרוזת. בהכרזת ה־DATA שממנה ייקראו ערכי המשתנים יופיעו מחרוזות במקומות המתאימים למשתנים המחרוזתיים.

דוגמה:

הוראת READ וההכרזה DATA ממנה יקראו הנתונים:

```
100 READ A,B,A$,X,R$
190 DATA 5,22,"EXAMPLE 1", 3, "NUMBER"
```

בזמן ביצוע הוראה 100 יקבלו המשתנים המספריים A, B ו־X את הערכים 22,5 ו־3 בהתאמה, ולמשתנים המחרוזתיים A$ ו־R$ יוצב בהתאמה התוכן EXAMPLE1 ו־NUMBER.

אם בהכרזת ה־DATA תופיע מחרוזת במקום שבו צריך להיות קבוע מספרי, תודיע התכנית על שגיאה וביצוע התכנית יופסק. למשל, אם במקום ההכרזה 190 בדוגמה לעיל תופיע ההכרזה

```
190 DATA 5, "EXAMPLE 2"
```

הרי שבזמן ביצוע ההוראה 100 תנסה התכנית להציב את המחרוזת EXAMPLE2 למשתנה המספרי B, וכתוצאה מכך תיווצר שגיאה והתכנית תדפיס את הודעת השגיאה ILLEGAL DATA.

PRINT

הטיפול במחרוזות בהוראת ה־PRINT הוסבר קודם. הטיפול במשתנים מחרוזתיים יוסבר להלן.

אפשר לרשום ברשימת הקבועים, המשתנים המספריים והמחרוזות המופיעים אחרי הוראת ה־PRINT גם משתנים מחרוזתיים, כדי שתוכנם יודפס בזמן ביצוע ההוראה.

דוגמה:

```
250 PRINT A,B, "THIS IS" ; A$
```

בעת ביצוע הוראה זו תודפס על גבי המסוף שורה, אשר יופיעו בה לפי הסדר הנתונים האלה: ערכו של המשתנה המספרי A, ערכו של המשתנה המספרי B, המלים THIS IS, ותוכנו של המשתנה המחרוזתי A$. למשל, אם ערכו של A הוא 5, ערכו של B 22, ותוכנו של A$ הוא EXAMPLE1 — תודפס השורה הזו על גבי המסוף:

```
5 22 THIS IS EXAMPLE 1
```

LET

הוראת ההצבה LET מאפשרת, בנוסף לטיפול בקבועים מספריים, משתנים מספריים וביטויים אריתמטיים כפי שתואר לעיל, גם להציב תוכן למשתנה מחרוזתי.

דוגמאות:

```
120 LET P$ = "MONTH"
310 LET S$ = H$
```

בשורה 120 תוצב המחרוזת MONTH למשתנה המחרוזתי P$. בשורה 310 תוצב המחרוזת המיוצגת על־ידי H$ אל המשתנה המחרוזתי S$.

INPUT

הוראה זו מאפשרת לתכנית לקבל נתונים מהמסוף כפי שהוסבר לעיל. המשתנים הרשומים אחרי הוראת ה־INPUT יכולים להיות משתנים מספריים או משתנים מחרוזתיים, ועל המשתמש בתכנית להקיש על גבי המסוף את ערכם של המשתנים בהתאם לדרישת התכנית.

דוגמה:

```
170 INPUT A,B$,A$,T
```

בהגיע התכנית להוראה 170 יופיע על המסוף סימן שאלה (?) המודיע למשתמש שהתכנית מצפה לנתונים. לפי הוראה זו על המשתמש להקיש ערך מספרי עבור המשתנה A, מחרוזת עבור המשתנה המחרוזתי B$, מחרוזת נוספת עבור המשתנה המחרוזתי A$ וערך מספרי עבור המשתנה המספרי T. הקשת הנתונים במסוף חייבת להיעשות בסדר המתאים לסדר הופעתם בהוראת ה־INPUT. בין כל שני נתונים חייב המשתמש להקיש פסיק (,). בזמן הקשת מחרוזת מספיק להפרידה מהנתונים האחרים על־ידי הקשת פסיקים לפניה ולאחריה, ואין הכרח להקיש גרשיים.

דוגמאות:

```
50 INPUT A$
```

בהגיע המחשב להוראה זו יופיע התו ? על המסוף, המודיע למשתמש שהתכנית מצפה לנתון. תשובת המשתמש, למשל, תהיה:

```
? YES
```

כלומר המשתמש יקיש את המחרוזת, ולאחר מכן ילחץ על מקש המשלוח (RETURN).

```
60 INPUT A$,H,H$,R$
```

כאשר תגיע התכנית לביצוע הוראה זו, יופיע התו ? על המסוף, ואז על המשתמש להקיש ארבעה נתונים: הראשון, השלישי והרביעי יהיו מחרוזות, והשני יהיה מספר. למשל —

```
? ONE,5,FIVE,EIGHT
```

אם התכנית מצפה לקבל מספר והמשתמש מקיש מחרוזת, תופיע הודעת השגיאה ILLEGAL DATA.

IF-THEN

בהוראה זו ניתן להשתמש במחרוזות ובמשתנים מחרוזתיים בדומה לשימוש שתואר לגבי קבועים מספריים, משתנים מספריים או ביטויים אריתמטיים, אך יש לזכור שהתנאי חייב להיות לגבי ערכים מספריים בלבד או לגבי מחרוזות בלבד, ולא ניתן לערב ביניהם.

דוגמאות:

```
80 IF A$ = "YES" THEN 300
110 IF P$ = Q$ THEN 40
190 IF H$< "POR" THEN 420
230 IF R$>=S$ THEN 200
```

בכל אחד מהמקרים לעיל תתבצע ההסתעפות רק אם התנאי המופיע בהוראה מתקיים, ולא — ימשיך ביצוע התכנית להוראה הבאה.

מערכים (ARRAYS)

מערך הוא טבלה של ערכים מספריים או מחרוזות, אשר הפנייה אליהם בתכנית נעשית על־ידי משתנה מספרי או משתנה מחרוזתי המאופיין על־ידי אינדקס.
דוגמה — מערך חד־ממדי (וקטור);
סדרת תאים, בכל תא ערך אחר. הפנייה לערך המשתנה הרצוי נעשית לפי מספר התא (האינדקס).

הערך הרשום בתוך התא	5	1	7	7	0	1	13	8	2
מספר התא	9	8	7	6	5	4	3	2	1

הערך של המשתנה "תא 3" הוא 13;
הערך של המשתנה "תא 7" הוא 7.

דוגמה: — מערך דו־ממדי (מטריצה)

סדרת תאים הבנויה משורות ועמודות; בכל תא ערך אחר.
הפנייה לערך המשתנה הרצוי נעשית לפי מספר השורה ולפי מספר העמודה (האינדקסים) המאפיינים את התא.

	מספר השורה							
	1	7	0	12	3	1	8	0
הערכים רשומים	2	2	0	0	3	4	5	2
בתוך התאים	3	10	1	0	0	13	1	15
	4	0	1	8	5	6	9	9
מספר העמודה		7	6	5	4	3	2	1

הערך של המשתנה "תא בשורה 2 עמודה 3" הוא 4;
הערך של המשתנה "תא בשורה 3 עמודה 5" הוא 0.

מערך קווי הנקרא וקטור (vector)
מערך קווי הוא למעשה סדרה בעלת אינדקס (subscript או index) אחד. לווקטור ניתן שם על־ידי שימוש בשם של משתנה מספרי או משתנה מחרוזתי (לפי העניין) ולאחריו (בסוגריים) האינדקס.

הצורה היא, לדוגמה:

A( )
A$( )

בתוך הסוגריים יכול לבוא ביטוי אריתמטי כלשהו. ערך ביטוי זה יקוטע למספר שלם שיהווה את האינדקס של הווקטור.
איברי הווקטור הם, לפי הסדר:

A(1),A(2),A(3),...,A(N),...

האיבר A(N) משמש כמשתנה מספרי לכל דבר ומותר להציב בו ערכים מספריים על-ידי הוראות הצבה וחישוב. כל הכללים החלים על משתנים מספריים נשמרים גם עבור איברי מערך מספרי. האיבר A$(N) משמש כמשתנה מחרוזתי לכל דבר, ומותר להציב בו מחרוזות ולהשתמש בו בכל ההוראות המתאימות למשתנים מחרוזתיים.

דוגמאות:

```
100 LET A(3) = 2*P+1
240 LET A$(6) = "XY"
340 LET A(2) = A(1) + A(4)
```

במקרה הראשון האיבר השלישי של הווקטור המספרי A יקבל את הערך שיחושב לביטוי 2P + 1.
במקרה השני, האיבר השישי בווקטור המחרוזתי A$ יקבל את התוכן XY.
במקרה השלישי, האיבר השני בווקטור המספרי A יקבל את הערך שהוא סכום המספרים הנמצאים באיבר הראשון ובאיבר הרביעי של אותו הווקטור.
האינדקס של הווקטור משמש אפוא ככתובת לתכניתן, המאפשרת לפנות לכל איבר בווקטור.

מערך מלבני הנקרא מטריצה (matrix)
מערך מלבני הוא "מלבן" של מקומות, שבו כל מקום מאופיין על-ידי שני אינדקסים:

אינדקס השורה	(row subscript)
אינדקס העמודה	(column subscript)

גם למטריצה ניתן שם המסומל על-ידי סמל משתנה מספרי או משתנה מחרוזתי (לפי העניין), ולאחריו בסוגריים יבואו שני אינדקסים כשביניהם פסיק: הראשון הוא אינדקס השורה והשני אינדקס העמודה. כמו בווקטור, יכולים האינדקסים להיות ביטויים אריתמטיים כלשהם.

הצורה של איבר המטריצה תהיה:

```
A$( , )
A ( , )
```

מלבן איברי המטריצה בעלת M שורות ו־N עמודות נראה כך:

```
A(1,1) A(1,2) A(1,3) ..... A(1,N)
A(2,1) A(2,2) A(2,3) ..... A(2,N)
•
•
•
•
A(M,1) A(M,2) A(M,3) ... A(M,N)
```

וקטור הוא מקרה פרטי של מטריצה בעלת עמודה אחת.

דוגמאות:

```
310 LET A(3,5) = 3*X+X**2
340 LET A(I+3,2*J) = 9
270 LET A(R,12) = A(R*3,10) * A(2,R+1)
200 IF A$(8,S+2) = B$(1,6) THEN 290
180 READ A$(1,2),A(1,2),A(1,3), B (2,1)
```

DIM

הכרזת הממד DIM מגדירה את ממדי המטריצות בתכנית (ממד המטריצה פירושו מספר השורות ומספר העמודות במטריצה).

צורת ההוראה:

לווקטור: <LN> DIM <V> (<N>)

למטריצה: <LN> DIM <V>(<M> ,<N>)

וכאן חייבים <N>, <M> להיות מספרים שלמים חיוביים. ניתן באותה שורה להכריז על ממדי מספר כלשהו של וקטורים ומטריצות.

למשל:

```
10 DIM X(100), Y(20,50), A$(5), Z$(2,2)
```

כל וקטור או מטריצה שנעשה בהם שימוש בתכנית חייבים להופיע בהוראת הממד, שבה נקבע מספר השורות ו/או העמודות המקסימלי שלהם.
ההכרזה יכולה לבוא בכל מקום בתכנית, אולם רצוי מאוד להכריז על כל הווקטורים והמטריצות בראש התכנית (בכל מקרה, לפני השימוש בהם בתכנית).
אם במהלך ביצוע התכנית יחרוג אחד האינדקסים של וקטור או של מטריצה

מהחסמים שלו: התחתון 1 והעליון כפי שנקבע בהכרזת ה־DIM, תופיע הודעת שגיאה בנוסח:

SUBSCRIPT OUT OF RANGE

וביצוע התכנית ייפסק.

במהלך התכנית ניתן לשנות את ממדי המטריצה, בתנאי שהממדים החדשים אינם גדולים מאלו המצויינים בהוראות ה־DIM.

קבצים

שפת ה־BASIC מאפשרת טיפול בקבצים. קיימים הבדלים ניכרים בשיטות הטיפול בקבצים בשפת ה־BASIC, בהתאם למחשב המסוים שבו נעשה שימוש, ואין כל אחידות בנושא זה.

הטיפול בקבצים מחייב ירידה לעומקו של עניין, דבר שהוא מעבר לייעודו של ספר זה.

מסיבות אלה לא מכיל פרק זה הוראות הקשורות בטיפול בקבצים. על הקורא המעוניין לפנות לחוברת ה־BASIC של המחשב שברשותו.

פרק 4.

טכנולוגיות עיבוד מידע

הפיתוח והיישום של מערכות מידע ממוחשבות מודרניות נשענים על שני ענפים ראשיים בטכנולוגיה — מסדי נתונים ותקשורת.
הראשון מאפשר שילוב נתונים; השני מקרב את המידע למשתמשים. גם בעבר היו קיימים בארגון קבצי נתונים וכלים להעברת נתונים, אולם הטכנולוגיות הנוכחיות מאפשרות הן קשר בין הקבצים הנמצאים במסד הנתונים והן תקשורת אלקטרונית בין מחשב למחשב ובין מחשבים להיתקני ציוד היקפי.

ארגון קבצים (FILE ORGANIZATION)

מחשבים מייצרים שני סוגי פלט — האחד לשימוש האדם והשני לשימוש המחשב. הסוג השני דורש שהפלט יירשם על אמצעי אחסון. כלומר יש ליצור מאגרי נתונים או קבצי נתונים שמאוחר יותר יוזנו לזיכרון המחשב ללא צורך בהקשה מחדש. אמצעי האחסנה המחזיקים קבצים נקראים יחידות אחסון חיצוני ונדונו בפרק 2. קיים מגוון של אמצעי אחסון לנתונים, של שיטות לארגון הנתונים על גבי יחידות האחסון ושל דרכי גישה אליהם.

הגדרות יסוד בארגון קבצים

סדרת ההגדרות שלהלן הולכת מלמטה למעלה, מהיישות האלמנטרית ביותר ליישויות מקיפות יותר.

תו (character) — יחידה אלמנטרית של נתונים המייצגת כל סמל אפשרי בשפה — אות (A ,B, C), ספרה (3, 0), וסימן מיוחד (?, $).

שדה (field) — מחרוזת של תווים המתארת תכונה של פריט, לדוגמה, מספר תעודת זהות.

רשומה (record) — קבוצת שדות בעלי קשר לוגי ביניהם, המתארת פריט, — רשומת תלמיד.

קובץ (file) — קבוצת רשומות השייכות לנושא מסוים, למשל — קובץ עובדים. בדרך־כלל כל הרשומות בקובץ הן בעלות תבנית זהה.

מסד נתונים (data base) — אוסף קבצים השייכים ליחידה ארגונית מסוימת, למשל — מסד נתונים פיננסי, מסד נתוני ייצור.

ההגדרה האחרונה בלתי מוגבלת. היא יכולה להתייחס למסד נתונים חשבונאי שמכיל את כל הקבצים החשבונאיים, אך גם למסד הנתונים של כל המפעל. כמו כן, ההגדרה מושגית ומימושה במציאות יכול לנוע מקבוצה של קבצים קונבנציונליים עד מערכת לניהול מסדי נתונים מודרנית.

בנוסף לעיל יש להגדיר מספר מושגים ספציפיים:

שדה מפתח (key field) — שדה ברשומה שבאמצעותו ניתן לזהותה. למשל מספר תעודת זהות ברשומת אזרח. בדרך־כלל ממיינים רשומות קובץ לפי הסדר של שדות המפתח שלהן.

קובץ אב (master file) — קובץ המכיל את כל הרשומות השייכות לנושא מסוים, רשומה אחת לכל שדה־מפתח. קבצי־אב מתארים את המצב (סטאטוס) של פריטי הנושא בזמן מסוים, למשל קובץ אב־מלאי יכיל את כל פריטי המלאי, רשומה לפריט, ויתאר את מצב המלאי של הפריטים ברגע מסוים. קבצי־אב הם קבצים קבועים ומעודכנים תקופתית.

קובץ תנועות (transaction file) — קובץ המכיל את הרישומים של תנועות פיסיות השייכות לנושא מסוים. למשל רישומי כניסות ויציאות של פריטים למחסן המלאי במשך יום עבודה. בקובץ תנועות יכולות להיות כמה רשומות בעלות שדה־מפתח זהה, למשל כאשר לפריט מלאי יש שתי יציאות מהמחסן באותו יום. קבצי התנועות הם זמניים ומשמשים לעדכון קבצי־אב.

קובץ היסטורי (historical file) — מהדורות קודמות (דורות ישנים) של קבצי־אב או קבצי־תנועות. קבצים היסטוריים נשמרים בדרך־כלל לצורכי גיבוי (back-up), לצורכי סטטיסטיקה ולצורכי בקרה וביקורת.

בעיבוד קבצים מעורבות שתי פעילויות עיקריות:

שליפה (retrieval) — השגת רשומה דרושה מתוך קובץ, למשל גליון ציונים של סטודנט מס' 12382.

עדכון (update) — שינוי קובץ אב ע"י תנועות שנצברו בקובץ תנועות.

המטרה העיקרית של כל יישום ממוחשב היא לספק בשעת השליפה את הנתונים הדרושים לצורכי המשתמש, בעוד שעדכון הוא רק אמצעי להשיג זאת. במלים אחרות, אין צורך לעדכן הקבצים אם לא חזויה דרישה לשליפה בעתיד.

תכונות של היתקני אחסון חיצוניים

יחידות האחסון החיצוני במערכת המחשב מתחלקות לשני סוגים, בהתאם לשיטת ארגון הקבצים המאוחסנים עליהן.

יחידות סדרתיות (serial devices) — ביחידות אלו הרשומות מסודרות זו אחר זו לפי סדר (עולה או יורד) של שדה־מפתח. קריאה או כתיבה של רשומות מחייבת מעבר סדרתי על הרשומות כולן עד לרשומה הרצויה.

יחידות סדרתיות הן כונן סרטים מגנטי, קוראת כרטיסים מנוקבים וקוראת סרט נייר מנוקב. יתרונן של יחידות אלה במחירן הזול יחסית.

יחידת גישה ישירה (direct access devices) — ביחידות אלו יכול המחשב להגיע ישירות לרשומה הנמצאת במקום כלשהו בקובץ ללא צורך בקריאת הרשומות הקודמות לה.

יחידת גישה ישירה נפוצה היא כונן תקליטים מגנטי.

יחידות אלו יקרות בהרבה מהיחידות הסדרתיות.

שיטות ארגון קבצים

ארגון הרשומות בקובץ צריך להיעשות בצורה שתאפשר עיבוד ושימוש יעילים בנתונים הנמצאים בקובץ. שיטת ארגון הקבצים המתאימה ליישום מסוים תלויה בתכונות הקבצים וביישומים שעבורם נדרשים הקבצים; יש לנתח גורמים אלה לצורך תיכון נכון של ארגון הקבצים. ארגון יעיל של הקבצים יכול להשפיע בצורה משמעותית על יעילות השליפה ועל העדכון של מערכת המידע הממוחשבת.

קיימים שלושה מבנים בסיסיים של ארגון קבצים:

א. קובץ סדרתי (sequential file)

ב. קובץ טבלתי (indexed sequential file)

ג. קובץ ישיר/אקראי (random/direct access file)

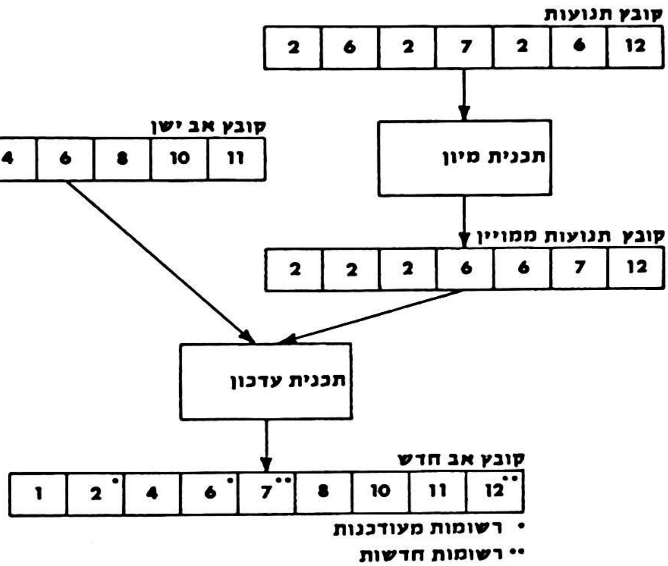
ארגון קבצים סדרתי

קבצים סדרתיים מאוחסנים בדרך־כלל על גבי סרטים מגנטיים, אך גם כרטיסים מנוקבים, סרט נייר מנוקב, תקליט מגנטי או תוף מגנטי יכולים לשמש לאחסון קבצים סדרתיים.

בשיטה זו מסודרות הרשומות בקובץ לפי שדה־מפתח בסדר עולה או יורד. בזמן עיבוד (קריאה, כתיבה, הוספה, ביטול או שינוי של רשומות בקובץ) של קובץ סדרתי, נעשה העיבוד לכל רשומה לפי סדר הופעתה בקובץ; אין אפשרות לבצע עיבוד לרשומה לפני שבוצע העיבוד הנדרש לכל הרשומות שלפניה. בעיבוד הדורש הוספה או ביטול של רשומות מקובץ סדרתי יש להעתיק את הקובץ מחדש: מאחר שהרשומות בקובץ הקיים כתובות אחת אחרי השנייה, אין מקום על גבי היתקן האחסון לרשומה נוספת בין שתי רשומות קיימות, ומאותה סיבה אין אפשרות לבטל רשומה הכתובה על גבי הקובץ מבלי להשאיר חלל ריק

במקום שבו הרשומה היתה כתובה. לכן, כאשר בזמן העיבוד יש לבטל או להוסיף רשומות לקובץ, נקרא הקובץ הקיים (הישן) לצורך קבלת הנתונים, וקובץ חדש נכתב — זה יכיל את הנתונים החדשים, יכלול את הרשומות החדשות שנוספו, ויצומצמו בו הרווחים שנוצרו עקב הרשומות שבוטלו.
מאחר שקריאה וכתיבה של הרשומות בקובץ הסדרתי נעשית לפי סדר הופעת הרשומות בקובץ האב, חייב גם עדכון הקובץ להיעשות בסדר זה, ולכן את כל קובץ התנועות — המכיל רשומות עם שינויים לקובץ — יש למיין באותו סדר מיון של קובץ האב שאותו יש לעדכן.

לדוגמה, מהלך עדכון של קובץ אב סדרתי (תרשים 4.1):



תרשים 4.1: עדכון קובץ סדרתי
(המספרים מייצגים שדות־מפתח של רשומות)

הקלט למהלך העדכון הוא קובץ אב וקובץ תנועות; שניהם ממויינים לפי אותו סדר מיון. הפלט ממהלך העדכון הוא קובץ אב חדש, שנעשו בו כל השינויים הדרושים, והוא ממויין בסדר המיון של הקבצים האחרים (קבצי הקלט).
תכנית העדכון קוראת רשומה אחת מכל אחד מהקבצים, ובודקת את שדות המפתח בשתי רשומות:

— אם מפתח הרשומה שנקראה מקובץ האב נמוך ממפתח הרשומה שנקראה מקובץ התנועות — פירוש הדבר שאין עדכון לרשומה שנקראה מקובץ־האב והיא מועתקת ללא שינוי אל קובץ האב החדש; אחרי כתיבת הרשומה בקובץ האב החדש, התכנית קוראת את הרשומה הבאה בקובץ האב, וחוזרת לבדיקת שדות המפתח.
— אם שדה המפתח של הרשומה שנקראה מקובץ האב זהה לשדה־המפתח של הרשומה שנקראה מקובץ התנועות — פירוש הדבר שיש לבצע עדכון לרשומה מקובץ האב בהתאם לנתונים שברשומה שנקראה מקובץ התנועות. לאחר עדכון הרשומה, נכתבת הרשומה המעודכנת על גבי קובץ האב החדש. כאשר התנועה שנקראה מקובץ התנועות פירושה ביטול הרשומה מקובץ האב, לא תכתב רשומת האב על גבי קובץ האב החדש. בגמר העדכון קוראת התכנית את הרשומה הבאה מכל אחד משני קבצי הקלט, וחוזרת לבדיקת שדות המפתח.
— אם שדה המפתח של הרשומה שנקראה מקובץ האב גדול משדה המפתח של הרשומה שנקראה מקובץ התנועות — פירוש הדבר שיש להוסיף רשומה חדשה לקובץ האב החדש, בהתאם לנתונים ברשומה שנקראה מקובץ התנועות. התכנית כותבת רשומה זו בקובץ האב החדש, ולאחר מכן קוראת את הרשומה הבאה מקובץ התנועות וחוזרת לבדיקת שדות המפתח.

מתיאור מהלך העדכון נובע שאם מספר התנועות קטן, חלק גדול מהכתיבה יהיה פעולת העתקה גרידא.

יתרונות ארגון קבצים סדרתי — חסכוני במקום אחסון, פשוט, וניתן להפעלה בהיתקני אחסון זולים. חסרונותיו — אין אפשרות לקרוא רשומה מסוימת מתוך הקובץ מבלי לקרוא את כל קודמותיה, ולכן העיבוד דורש זמן ניכר. אין אפשרות להוסיף או לבטל רשומות מקובץ קיים מבלי להעתיק אותו, ולכן במהלך העדכון יש לכתוב מחדש את הקובץ. יש למיין את קובץ התנועות לפי סדר המיון של קובץ האב לפני עדכון קובץ־האב.

ארגון קבצים טבלתי

קבצים מסוג זה מאוחסנים על גבי תקליטים מגנטיים. בזמן בניית קובץ טבלתי, ממויינות הרשומות בסדר עולה לפי שדה המפתח, והן נכתבות על גבי התקליט בסדר זה. בתחילת התקליט מצוי מדריך הגלילים (cylinder index) המכיל את שדה המפתח של הרשומה הגבוהה ביותר שנמצאת על כל גליל של התקליט. בנוסף לטבלה זו, נמצא במשטח העליון של כל גליל מדריך המסילות (track index) המכיל את שדה המפתח של הרשומה הגבוהה ביותר שנמצאת על כל משטח באותו גליל.

בזמן עדכון קובץ האב, מתבצע חיפוש רשומה מסוימת, ששדה המפתח שלה נתון, לפי השלבים האלה:

1. תכנית המחשב סורקת את מדריך הגלילים, ומוצאת באיזה גליל כתובה הרשומה הרצויה.
2. כשנמצא הגליל שבו מופיעה הרשומה הרצויה, סורקת התכנית את מדריך המשטחים של אותו גליל, ומוצאת באיזה משטח בגליל כתובה הרשומה הרצויה. באופן זה מגיעה התכנית למסילה שבה נמצאת הרשומה.
3. התכנית סורקת את המסילה, ומוצאת בה את הרשומה הרצויה.

אם לצורך עדכון רשומה יש לשנות את הנתונים ברשומה — נכתבים הנתונים החדשים באותו המקום שבו נמצאת הרשומה בקובץ. בעיית ביטול רשומות קיימות והוספת רשומות חדשות לקובץ מסובכת יותר, כיון שהרשומות מסודרות זו אחר זו ללא רווחים ביניהן, כמו בקובץ המאורגן סדרתית.

ביטול רשומה מקובץ האב נעשה על־ידי סימון הרשומה כמבוטלת, והשארתה במקומה בקובץ עד לארגון מחדש של הקובץ (ראה להלן).

הוספת רשומה חדשה לקובץ נעשית באופן הזה: התכנית מחפשת בעזרת מדריך הגלילים ומדריך המסילות את המקום שבו צריכה הרשומה החדשה להיות על גבי הקובץ, בהתאם לשדה המפתח של הרשומה וסדר מיון הקובץ. אם במסילה שנמצאה קיים מקום פנוי המספיק לכתיבת הרשומה החדשה — התכנית תזיז את הרשומות הנמצאות על אותה מסילה, והן בעלות שדה מפתח גבוה יותר, כדי לפנות את המקום המיועד לרשומה החדשה, ואז תכתוב אותה במקומה, כך שמיקום הרשומות הישנות והרשומה החדשה יהיה בהתאם לסדר המיון של הקובץ.

אם במסילה שנמצאה אין מקום פנוי עבור הרשומה החדשה, התכנית תזיז את הרשומות בעלות המפתח הגבוה יותר, הנמצאות על אותה מסילה, כדי לפנות את המקום המיועד לרשומה החדשה. כתוצאה מהזזה זו, הרשומה האחרונה על גבי המסילה תהיה מחוסרת מקום על המסילה. במקום שפונה עקב הזזת הרשומות נכתבת הרשומה החדשה. הרשומה האחרונה, שהוצאה מהמסילה, תיכתב על־ידי התכנית על גבי התקליט בשטח מיוחד הנקרא — שטח גלישה.

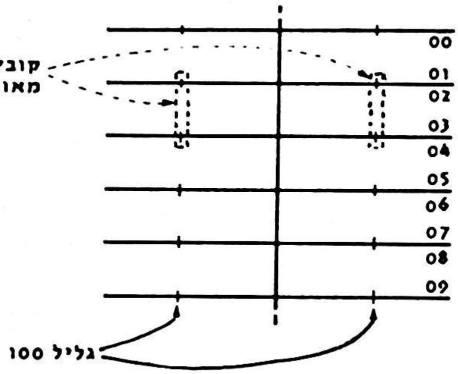
כאשר התכנית מחפשת רשומה מסוימת — לאחר סריקת המסילה שנמצאה לפי מדריך הגלילים ומדריך המסילות — סורקת התכנית גם את שטח הגלישה של אותה מסילה, כאילו היה שטח הגלישה המשך של המסילה.

בדרך זו נעשה לכאורה מיקום הרשומות על גבי הקובץ לפי סדר המיון של הקובץ, בעוד שלמעשה מחולקות הרשומות בין שני שטחי הקובץ — השטח הראשי (prime data area) ושטח הגלישה (overflow area).

לאחר כמה עיבודים, שבעקבותיהם נוספו רשומות רבות לקובץ — ייתכן שמספר הרשומות בשטח הגלישה יהיה גדול, דבר שיגזול זמן בחיפוש אחר רשומה בקובץ. לכן מפעם לפעם מאורגן הקובץ מחדש. הארגון מחדש פירושו מיון כל הרשומות השייכות לקובץ (הרשומות הנמצאות בשטח הראשי והרשומות הנמצאות

בשטח הגלישה) לפי סדר המיון של הקובץ, סילוק הרשומות שסומנו כמבוטלות, וכתיבת הקובץ מחדש. כתוצאה מהארגון מחדש יהיו כל הרשומות מסודרות בשטח הראשי לפי סדר המיון, שטח הגלישה יהיה ריק ומוכן לקלוט רשומות חדשות, ומדריכי הגלילים והמסילות יהיו מעודכנים בהתאם לסדר החדש בקובץ.

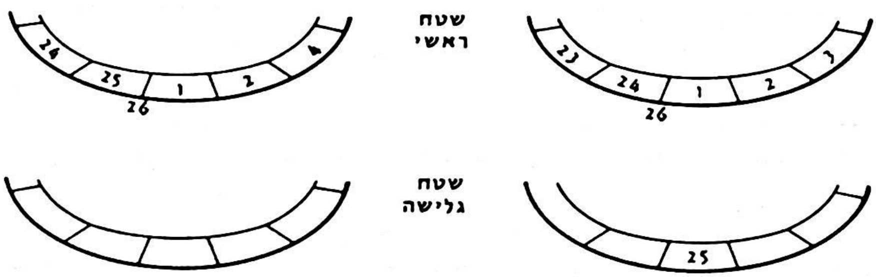
דוגמה: קובץ אב של עובדים במפעל מכיל רשומה אחת לכל עובד, הכוללת את מספרו, שמו ופרטים אחרים של העובד. הקובץ ממויין סדרתית בסדר עולה לפי מספר העובד (שדה מפתח) ומאוחסן על גבי תקליט. השטח הראשי של הקובץ נמצא בגליל 100 במשטחים 1, 2, 3, ו־4 (תרשים 4.2)



תרשים 4.2: קובץ אב מאוחסן בשטח ראשי

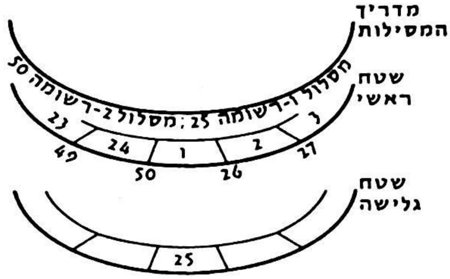
בתרשים 4.3א ניתן לראות את תוכן השטח הראשי ואת תוכן שטח הגלישה כאשר הקובץ נוצר. תרשים 4.3ב מדגים את השימוש בשטח הגלישה כתוצאה מהכנסת רשומה חדשה של עובד מספר 3.

בדוגמה זו הועברה רשומה 25 לשטח הגלישה.



תרשים 4.3א׳, ב׳: מבנה הקובץ
לפני ואחרי הוספת רשומה 3.

תרשים 4.4 מראה את מדריך המסילות של קובץ העובדים המתואר בדוגמה. כזכור, מדריך המסילות שומר את שדה המפתח של הרשומה הגבוהה ביותר בכל מסילה.



תרשים 4.4 : מדריך המסילות

בדוגמה זו, כדי לקרוא את רשומה 4 למשל, קוראת התכנית את מדריך הגלילים, המפנה אותה לגליל 100. בגליל 100 קוראת התכנית את מדריך המסילות, המראה ששדה המפתח הגבוה ביותר במסילה 1 הוא 25 (רשומה 25 נמצאת בשטח הגלישה). כיון שרשומה 4 הנה בעלת שדה מפתח הקטן מרשומה 25, יודעת התכנית שרשומה 4 נמצאת על מסילה 1, ולפיכך תקרא התכנית את הרשומות ממסילה 1, שביניהן תמצא את רשומה 4.

כדי לקרוא את רשומה 27, קוראת התכנית את מדריך הגלילים ואת מדריך המסילות. על־ידי בדיקת מדריך המסילות, יודעת התכנית ששדה המפתח הגבוה במסילה 1 הוא 25, ולכן רשומה 27 לא תימצא על מסילה 1. שדה המפתח הגבוה במסילה 2 הוא 50, ולכן רשומה 27 תימצא במסילה 2.

יתרונה העיקרי של שיטת הארגון הטבלתי הוא שניתן לגשת לכל רשומה בקובץ בצורה כמעט ישירה. מכאן נובע יתרון אחר של שיטה זו, והוא שאין צורך למיין את התנועות הבאות לעדכן את הקובץ.

יתרונות אחרים של השיטה — א. הוספת רשומות לקובץ איננה מחייבת כתיבת הקובץ מחדש; ב. גישה מהירה וישירה לנתונים בקובץ; תכונות אלה מאפשרות עיבוד מקוון של קבצים אלו.

יתרון נוסף של השיטה הוא שאפשר לעבוד עם קבצים טבלתיים כאילו היו קבצים סדרתיים, דבר שהוא נוח מאוד לצרכי העדכון של קובץ האב.

חסרונותיה של השיטה — התכנות מסובך יותר מבשיטת הקבצים הסדרתיים, הצורך לארגן מחדש את הקבצים מפעם לפעם, והימצאותן של הרשומות המבוטלות על גבי הקובץ עד לארגון מחדש, דבר הגורם לבזבוז במקום לאחסון. בזבוז נוסף נובע מהצורך בשמירת מקום לשטח הגלישה. חסרון אחר — עדכון הקובץ נעשה על גבי הקובץ הקיים (כלומר — ללא העתקתו), וכתוצאה מכך

לא נוצר אוטומטית קובץ היסטורי המשמש גיבוי לקובץ החדש. קבצים טבלתיים מחייבים שימוש בתקליטים ובכונני תקליטים מגנטיים שהם אמצעי אחסון יקרים יחסית.

ארגון קבצים ישיר/אקראי

בשיטת הארגון הישיר מאוחסנים הקבצים על גבי תקליט מגנטי. בשיטה זו לכל רשומה בקובץ כתובת פיסית הקובעת את מיקומה על גבי התקליט — הכתובת הפיסית של כל רשומה מבוססת על שדה המפתח של הרשומה, אך איננה זהה לו, ולכן הרשומות אינן בהכרח מסודרות לפי סדר המיון של שדות המפתח. חשוב להדגיש שמיקום הרשומה על גבי התקליט ניתן לקביעה באמצעות נוסחה המבוססת על שדה המפתח של הרשומה. שדה זה מהווה את הכתובת "הלוגית" של הרשומה.

בזמן עיבוד קובץ ישיר, כדי למצוא רשומה מסוימת ששדה המפתח שלה נתון, מתרגמת תכנית המחשב בעזרת הנוסחה הנתונה את שדה המפתח של הרשומה לכתובת הפיסית על התקליט, ובעזרת כתובת זו יכולה התכנית להגיע מיידית לרשומה הרצויה.

שינוי נתונים ברשומת קובץ האב נעשה על־ידי כתיבת הנתונים החדשים ברשומה הקיימת. ביטול רשומה נעשה על־ידי סימונה כמבוטלת, והוספת רשומה נעשה על־ידי כתיבת הרשומה החדשה במקומה בקובץ, בהתאם לשדה המפתח שלה ותרגומו לכתובת הפיסית בעזרת הנוסחה.

בשיטה זו קיים בזבוז ניכר של מקומות בקובץ, מאחר שנשמר מקום על גבי התקליט לכל רשומה אפשרית — אם היא בשימוש ואם לא. קיימות שיטות מתוחכמות לחישוב הכתובות הפיסיות מתוך שדה המפתח, הגורמות שמספר המקומות המבוזבזים על גבי התקליט יהיה מינימלי, אלא שאז קיים סיכון שנוסחה כזו תיתן כתובת זהה לשני שדות מפתח שונים; קיימות שיטות להתגבר גם על בעיה זו.

דוגמה: בקובץ עובדים במפעל שדה המפתח של הרשומה הוא מספר העובד. מספרי העובדים במפעל נמצאים בין 500 ל־999 (אלו הן כתובות לוגיות).

שיטה אפשרית לקביעת כתובת פיסית של כל רשומה על גבי התקליט היא השיטה הזו:

הספרה השמאלית במספר העובד תציין את מספר הגליל; הספרה האמצעית תציין את מספר המשטח בגליל, והספרה הימנית במספר העובד תציין את מקום הרשומה במסילה. למשל, הרשומה המתאימה לעובד שמספרו 500 תימצא בגליל 5, במשטח 0, והיא תהיה הרשומה הראשונה (שמספרה 0) על גבי המסילה. הרשומה של עובד 501 תמצא אף היא בגליל 5, במשטח 0, ותהיה הרשומה השנייה (שמספרה 1) על־גבי המסילה. הרשומה המתאימה לעובד 728 תמצא בגליל 7 במשטח 2, במקום התשיעי במסילה.

הנוסחה לקביעת הכתובות הפיסיות המוצגת בדוגמה איננה השיטה היחידה האפשרית, אך מבין השיטות היא הפשוטה והבזבזנית ביותר; אם במפעל יש למשל רק 200 עובדים (שמספריהם, כאמור, בין 500 ל־999), הרי שחלק ניכר מהמקום שהוקצה לקובץ יישאר ריק: אם אין עובד שמספרו 659, הרי שהמקום העשירי במשטח 5 בגליל 6 יהיה ריק. אם הרשומה של כל עובד קצרה, כך שניתן למשל להכניס 30 רשומות לכל מסילה, השיטה המוצגת לא תנצל את אורכן הקצר של הרשומות, ולא תאפשר לכתוב יותר מ־10 רשומות במסילה. בשיטה זו אין גם אפשרות לבחור את המקום על גבי התקליט שבו יימצא הקובץ — הנוסחה קובעת שהקובץ ישתרע על גלילים 5, 6, 7, 8, ו־9. אם גלילים אלה תפוסים על־ידי קובץ אחר, יש צורך לקבוע נוסחה מתוחכמת יותר, אשר תמקם את הקובץ במקום הפנוי על גבי התקליט.

כאמור, קיימות שיטות מתוחכמות לקביעת נוסחאות לתרגום שדה המפתח של רשומה לכתובת פיסית על גבי התקליט. נוסחאות אלה מתחשבות בגודל הרשומות, במספר הרשומות שאפשר לכתוב על כל מסילה, במספר שדות המפתח שבהם יעשה שימוש מתוך שדות המפתח האפשריים, ובמקומות הפנויים על גבי התקליט המגנטי.

יתרונות ארגון הקבצים הישיר — גישה ישירה ומהירה מאוד לכל רשומה בקובץ. בגלל אפשרות הגישה הישירה אין צורך למיין את קובץ התנועות הבא לעדכן את קובץ האב. הוספת רשומות לקובץ וביטול רשומות מהקובץ הם פשוטים ואינם דורשים העתקה או ארגון מחדש של הקובץ. תכונות אלה של הארגון הישיר מאפשרות לעבוד ביעילות מרבית עם קבצים אלה בשיטת העיבוד המקוון. חסרונותיה של השיטה — הישארות שטחים ריקים על גבי התקליט המגנטי (דבר הגורם לניצולת נמוכה של הציוד), תכנות מסובך, וחסרונו של קובץ היסטורי לגיבוי הקובץ המעודכן, כתוצאה מעדכון הקובץ ללא העתקתו.

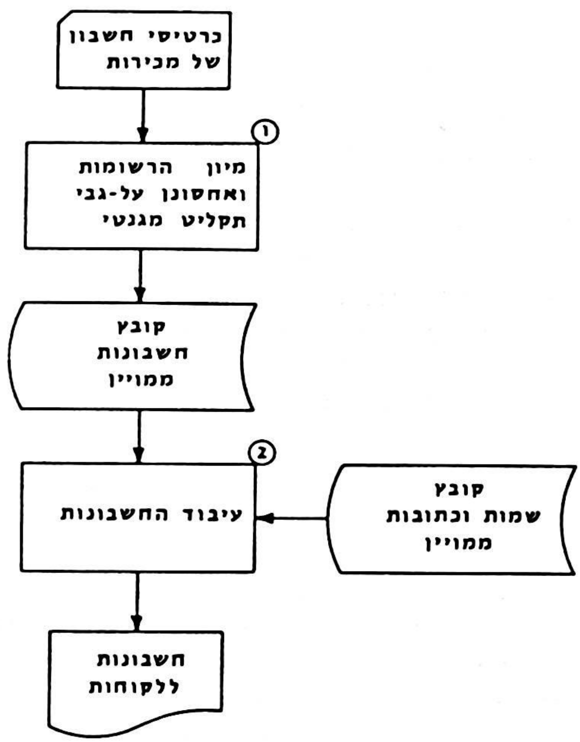
בחירה בשיטת הארגון המתאימה

בחירה נכונה בשיטת ארגון הקובץ יכולה להשפיע בצורה משמעותית על יעילות כל מערכת המחשב. הבעיה היא שאין כללים חד־משמעיים לקביעת שיטת ארגון הקובץ בכל יישום ויישום אפשרי.

ברוב המקרים מתאימות כמה שיטות ארגון לביצוע יישום מסוים, והבחירה בין השיטות האפשריות חייבת להתבסס על מספר גורמים. השיקולים המשפיעים על בחירה זו הם טכניים וכלכליים ולעתים יכולים להיות מנוגדים.
להלן דוגמה שבה אותו העיבוד מתבצע בשתי שיטות שונות של ארגון קבצים. בדוגמה זו כרטיסי המכירות של מפעל הם קלט לעיבוד עם קובץ אב הלקוחות. תכנית העיבוד קוראת את כרטיסי החשבונות, מחפשת את הלקוח בקובץ אב הלקוחות, ומכינה חשבון עבור הלקוח. כרטיסי החשבון המשמשים קלט לתכנית אינם ממויינים.

כאשר קובץ אב לקוחות הוא סדרתי — קיימים שני שלבים בתכנית (תרשים 4.5):

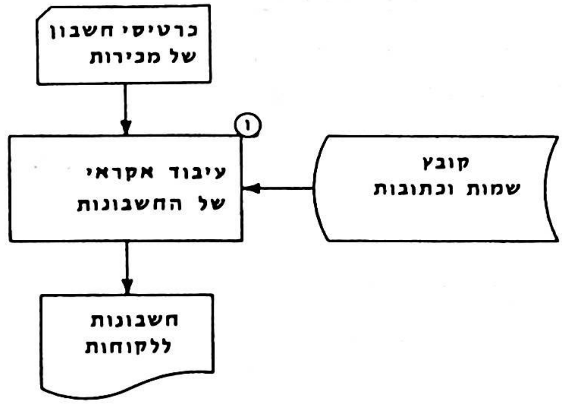
1. התכנית קוראת את כרטיסי החשבון, ממיינת אותם בהתאם למיון קובץ אב לקוחות, ומאחסנת אותם על גבי תקליט מגנטי במבנה של קובץ סדרתי.
2. התכנית קוראת את הרשומות מקובץ החשבונות הממויין ומקובץ אב לקוחות המכיל שמות וכתובות (תהליך הקריאה והשוואת שדות המפתח מוסברים בסעיף הדן בקבצים סדרתיים). התכנית מחשבת את הסכומים לתשלום, מדפיסה את החשבון ללקוח, וממשיכה לקרוא את הרשומה הבאה בשני הקבצים.



תרשים 4.5: עיבוד והפקת חשבונות — קובץ סדרתי

בקובץ אב לקוחות שהוא קובץ טבלתי או ישיר/אקראי, קיים רק שלב אחד בעיבוד (תרשים 4.6) — התכנית קוראת את כרטיסי החשבון, מחפשת את הרשומה המתאימה ללקוח בקובץ אב לקוחות (תהליך חיפוש הרשומה תואר בסעיפים הדנים בקובץ טבלתי ובקובץ ישיר/אקראי), וכאשר הרשומה נמצאת,

מחשבת התכנית את הסכומים לתשלום, מדפיסה את החשבון ללקוח, ולאחר מכן ממשיכה לקרוא את כרטיס החשבון הבא.



תרשים 4.6: הכנת חשבונות — קובץ טבלתי או ישיר/אקראי

בקובץ אב לקוחות שהוא טבלתי או ישיר/אקראי אין צורך למיין את כרטיסי החשבונות, והעיבוד כולו מבוצע בצעד אחד, כך שמתקיים יתרון ברור לעומת הקובץ הסדרתי. אך קיימים שיקולים נוספים בבחירת שיטת ארגון הקובץ, ולכן אי־אפשר לקבוע שארגון טבלתי או ישיר/אקראי יעילים יותר תמיד.

השיקולים לבחירת שיטת ארגון קבצים

פעילות (activity) הקובץ

מספר או אחוז הרשומות המעורבות בתהליך העיבוד. יתכן קובץ שבו הפעילות שווה בממוצע לכל הרשומות; יתכן קובץ אחר שבו רוב הפעילות נוגעת למספר קטן של רשומות. לדוגמה, 10 אחוז פעילות פירושם שבמשך תקופה מסוימת (או במהלך עיבוד מסוים) עודכנו או נקראו 10 אחוז מהרשומות בקובץ. ככל שגדל אחוז הפעילות כן כדאי להשתמש בקובץ סדרתי, ולהיפך — אחוז פעילות נמוך עושה את שיטות הארגון הטבלתי או הישיר לכדאיות יותר. אם יש לעדכן 50 רשומות בקובץ של 10,000 רשומות, לא תהיה השיטה הסדרתית יעילה, כיון שתחייב סריקה של כל 10,000 הרשומות.

תנועת (volatility) הקובץ

תכיפות הביטול וההוספה של רשומות לקובץ. לדוגמה, קובץ אב של מלאי חלפים במוסך הוא בעל תנועה רבה, ואילו קובץ אב חברי־כנסת הוא בעל תנועה נמוכה. יש לזכור שהוספה וביטול של רשומות בקובץ המאורגן סדרתי מחייבים

יצירת קובץ חדש, בעוד שבקובץ המאורגן טבלתי או בקובץ ישיר ניתן לטפל בתוספות ובביטולים ללא העתקת הקובץ.

רמת העדכניות

משך הזמן המקסימלי אשר עובר בין התנועה שחלה במציאות לבין עדכון קובץ האב כתוצאה מהתנועה. העדכניות הדרושה מוכתבת על־ידי המשתמש בקובץ האב.

זמן תגובה

יתרונן החשוב של מערכות המידע המודרניות הוא באפשרות לגישה מהירה וישירה לרשומות, בעיקר בשיטת העיבוד המקוון. בשיטת עיבוד כזו, המאפשרת זמן תגובה מהיר, הכרחי לעבוד עם קבצים המאפשרים גישה ישירה לרשומות, ואפשרות זו קיימת בקבצים המאורגנים באופן טבלתי או ישיר בלבד.

גודל הקובץ ופוטנציאל הגידול שלו בעתיד

המקום הדרוש לאחסון הקובץ, הנקבע על ידי מספר התווים בכל רשומה ומספר הרשומות בקובץ; קבצים המאורגנים סדרתי יכולים להיות בכל גודל שהוא. קבצים המאורגנים באופן טבלתי או ישיר מוגבלים לגודל היתקן האחסון, ומהירות הגישה לרשומות שבהם מושפעת ישירות מגודלם.

עלות האחסון

קבצים המאורגנים סדרתי ניתנים לאחסון על גבי היתקני אחסון זולים (סרט מגנטי, סרט נייר מנוקב, כרטיסים מנוקבים), בעוד שקבצים המאורגנים באופן טבלתי או ישיר דורשים היתקני אחסון יקרים (תקליט מגנטי).

בטיחות וגיבוי

נוהלי הבטיחות והגיבוי של קבצים המאורגנים סדרתי פשוטים וזולים לעומת קבצים המאורגנים טבלתי או ישיר. בקבצים המאורגנים סדרתי מתממש הגיבוי אוטומטית, ללא עלות נוספת, במהלך עדכון של קובץ אב.

מסדי נתונים

בתחילת עידן המחשב מוחשבו מערכות המידע בארגונים בעיקר כדי להקל על עומס הפעולות הידניות האופייניות למערכות לעיבוד תנועות (ראה פרק 6 להלן). בשנות השישים חדרה ההכרה שנתונים הנקלטים על־ידי כמה מערכות בדידות חייבים להיות משולבים אם הארגון מעוניין שמערכות המידע שלו תתמוכנה בקבלת החלטות ניהוליות.

לדוגמה, נניח שהנהלת ארגון תעשייתי שוקלת תוספת של אתר ייצור חדש. הגורמים שיש לקחת בחשבון נוגעים למכירות, עלויות השיווק, עלויות ייצור,

עלויות רכש, עלויות תובלה, כישורי כוח־אדם וכד׳. המידע הדרוש חייב להיאסף ממספר מקורות מידע, כאשר כל מקור מידע נשען על מערכת מידע נפרדת המספלת בעיבוד תנועות רוטיניות. אם מערכות אלו הן בדידות, ולא שולבו מלכתחילה, הרי שילוב הנתונים עלול להיות יקר מאוד ואולי בלתי־אפשרי.
בעיות מסוג זה יצרו את הצורך בפיתוח תוכנה לניהול מסדי נתונים, שתהיה מבוססת על העקרונות האלה:

אי־תלות בנתונים
תכניות יישומיות אינן צריכות להיות מושפעות משיטת הארגון של הקבצים. כל שינוי במבנה מסד הנתונים לא יגרום לשינויים בתכניות היישום. למעשה, המשתמשים ומתכנתי היישומים לא חייבים להכיר את התבנית הפיסית של הקבצים. מערכת מסד הנתונים צריכה גם להימנע מיתרות נתונים (data redundancy), כדי לחסוך בעדכונים של נתונים זהים שפוזרו בין קבצים שונים.

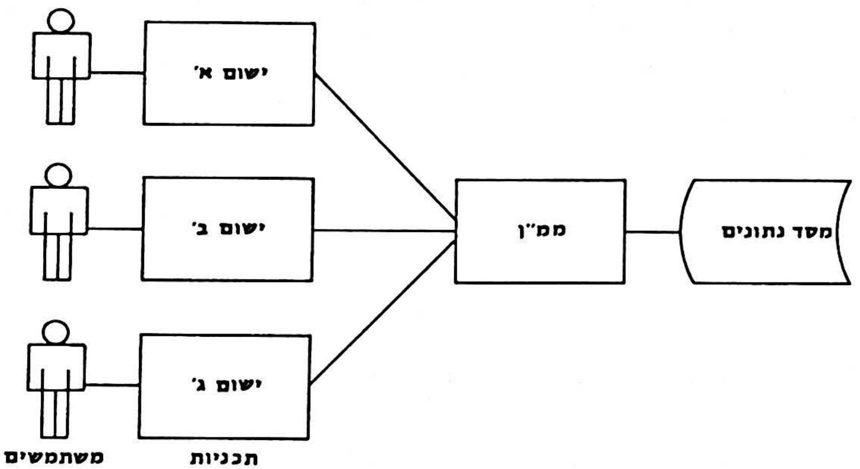
אי־תלות של תכניות
מערכת מסד הנתונים חייבת לשרת תכניות מחשב שנכתבו בשפות מחשב שונות. דבר זה יאפשר למתכנתי יישומים לבחור בשפת התכנות המתאימה ביותר לדרישות היישום, ללא קשר לתוכנת מסד הנתונים ולמבנה מסד הנתונים.

אי־תלות במערכת ההפעלה
מערכת מסד הנתונים לא תהיה תלויה במערכת ההפעלה של המחשב. במלים אחרות, תוכנת מסד הנתונים תתאים לגירסאות שונות של מערכות הפעלה ותוכנה בסיסית.

אי־תלות בחומרה
מערכת מסד הנתונים תהיה תואמת לתצורות חומרה שונות, גם אם הן מסופקות ע״י יצרני חומרה שונים. אידיאלית, מערכת מסד הנתונים צריכה לפעול עם כל סוג של חומרה שעומד באילוצי תצורה מינימליים, ללא צורך לעבור תהליכי הסבה ניכרים.

התוכנה המנהלת ומאחזקת מסדי נתונים נקראת מערכת מסד נתונים (data base system) — ממ״ן (DBS). זהו מוצר תוכנה בסיסית מורכב המכיל טבלאות רבות ושגרות תכנות. פיתוח הממ״ן ואחזקתה, דורשים מיומנות גבוהה מאוד. ממ״ן ניתנת לרכישה מיצרני ציוד ומבתי תוכנה.
ממ״ן מהווה ממשק בין שתי ישויות: משתמשים ונתונים (ראה תרשים 4.7). המשתמשים, המיוצגים ע״י תכניות יישומיות, רוצים לשלוף נתונים לצורך קבלת החלטות. הנתונים נמצאים על גבי התקני אחסון חיצוניים בצורה שאינה מובנת

למשתמש. המשתמש והמתכנת היישומי אינם מעוניינים במבנה הפיסי של הנתונים; הם מעוניינים רק במכשיר שליפה יעיל. מכשיר זה הוא ממ"ן. הממ"ן מתרגמת את דרישות תכנית היישום למפרטים טכניים שמסייעים בשליפת הנתונים הדרושים ובהצגתם למשתמש. כמובן שהממ"ן חייבת לדעת איך לבנות, לאחזק ולעדכן את מסד הנתונים. תפקיד הממשק מבוצע באמצעות טבלאות, מחונים ותכניות.



תרשים 4.7: ממ"ן כממשק בין משתמשים ונתונים

המבנה הפיסי של מסד הנתונים חייב להתאים לדרישות המידע השכיחות של המשתמשים ולספק לתכניות היישומיות שלהם את הנתונים המבוקשים. קיימות מספר גישות עיקריות למבנות הממ"ן:

הגישה ההיררכית (hierachical, tree) מציגה רשימת נתונים באמצעות מערך עץ. ניתן לגשת לנתונים דרך ענפי העץ, מלמעלה למטה. כל שדה יוצר קבוצה של שדות ברמה שמתחתיו, שיכולים ליצור רמות נמוכות יותר של שדות כפיפים וכן הלאה. ניתן לעבור ממבנה לוגי אחד (כלומר מבנה הנתונים הרצוי למשתמש אחד) למשנהו על־ידי שימוש במחונים, טבלאות ותוכנה, ללא צורך ברה־ארגון או בהכפלה של הנתונים הפיסיים. מבנה העץ הוא פשוט יחסית ונוח ליישומים העוסקים באנשים, במשפחות וכד' (למשל, יישומי כוח־אדם, סטודנטים).

גישת הרשת (network) מציגה את מסד הנתונים כרשת בעלת צמתים וערוצים. הצמתים מייצגים רשומות והערוצים מייצגים יחסים בין הרשומות.

בגישה זו השימוש שכיח כאשר בין הרשומות יש יחסי גומלין מרובים, כגון ביישומים העוסקים בהקצאת מכונות לתהליכי ייצור, בפיקוח על הייצור ועל המלאי בתהליכים המורכבים מחלקים רבים וכד׳.

הגישה היחסית (relational) כללית יותר מהגישות הקודמות. גישה זו מציגה את הנתונים כקבוצה של טבלאות המקיימות יחסי גומלין ביניהן באמצעות אופיינים משותפים. ניתן ליצור טבלאות חדשות מתוך הטבלאות הקיימות ובכך לספק את הדרישות לנתונים מצד התכניות היישומיות, בעזרת מגוון פעולות שניתן לבצע על הטבלאות. הגישה היחסית מכילה את התכונות העיקריות של גישת העץ וגישת הרשת ומאפשרת שליפה גמישה יותר במחיר של יתרות נתונים ויעילות נמוכה יותר.

מערכות רבות לניהול מסדי נתונים ניתנות להשגה מבתי תוכנה. הקריטריון העיקרי להערכתן הוא מידת התאמתן ליישומי המשתמש. כמו כן, יש לקחת בחשבון גם גורמים טכניים רבים, כמו בטיחות נתונים, אופן טיפולן בעדכון, יעילות השליפה, נוחות אחזקתן וביצועיהן.

תקשורת (COMMUNICATIONS)

המרכיבים העיקריים של כל תהליך תקשורת הם מקור ההודעה, האמצעי שבעזרתו עוברת ההודעה והמקבל. התהליך הפשוט ביותר הוא כאשר אדם אחד מדבר אל זולתו. האדם המדבר הוא למעשה משדר ההודעה או המקור, האוויר משמש כאמצעי להעברת ההודעה והאדם השומע משמש כמקבל ההודעה. הדבר נכון ביחס לכל צורות התקשורת הקיימות, כגון מערכת הטלפון, הרדיו והטלוויזיה. כל השיטות משתמשות בשלושת המרכיבים אשר צויינו לעיל.

כדי ליצור קשר חייב מידע כלשהו להיות מועבר. ״מידע״ בתורת התקשורת הוא כל סימן מוגדר. הוא יכול להיות סדרת אותיות, קבוצה של טונים מוסיקליים, קבוצת תמונות וכד׳. מידע זה מהווה הודעה (message).

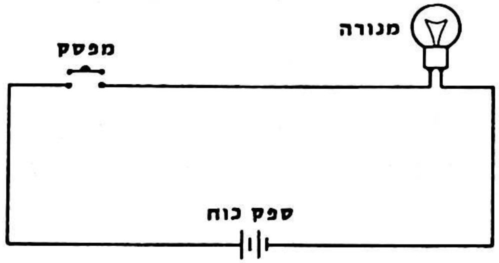
כדי שהתקשורת תהיה יעילה היא חייבת להיות מובנת. במערכת התקשורת הפשוטה נשלחת ההודעה בכיוון אחד בלי כל ידיעה אם המקבל מבין אותה או אינו מבינה. לדוגמה, אם אדם המקבל הודעה באמצעות הטלפון אינו מבין חלק מההודעה, הוא יבקש לחזור על החלק הבלתי־מובן. אם חזרה זו אינה ניתנת לביצוע, הרי שההודעה שתתקבל לא תהיה מלאה, או תובן שלא כהלכה.

האמצעי המשמש להעברת המידע הוא אמצעי מוגבל. לדוגמה, ספר זה מוגבל במספר עמודיו ובגודלו. האמצעי המוגבל להעברת המידע נקרא ערוץ (channel); בשיחה בין בני־אדם ערוץ התקשורת הוא האוויר, במערכת

הטלפונים ערוץ התקשורת הוא מוליך החשמל, ובשידור רדיו או בטלוויזיה הערוץ הוא תחום תדירות מסוים.
רעש (noise) הוא גורם חשוב בתהליך התקשורת. כרעש מוגדר כל גורם אשר מפריע להעברת המידע בצורה נקייה ונכונה. אנו נתקלים ברעשים כאלה יום יום, אם במערכת הטלפונים, כאשר נשמעות שיחות זרות או רעשים שונים הפוגמים באיכות השיחה, ואם על מסך הטלוויזיה כאשר נוצר "שלג" עקב רעשים המפריעים לקליטה תקינה וכדומה.

מערכות תקשורת

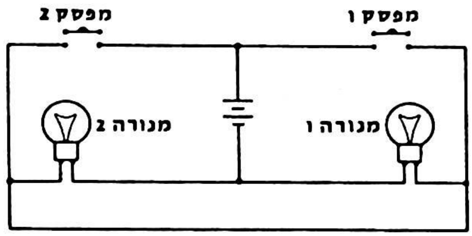
המאפיין העיקרי במערכות תקשורת הוא הזמן הדרוש להעברת המידע. מערכות זמן אמתי הן מערכות תקשורת אשר בהן מתקבלת ההודעה בזמן האפשרי הקצר ביותר; אדם המקשיב לאדם שני בשיחת טלפון מקבל את ההודעה בזמן אמתי. כלומר הוא מקבל אותה ללא השהיות. כאשר נשלחת ההודעה במכתב, אין היא מתקבלת מיד. כלומר מערכות כאלו אינן פועלות בזמן אמתי.
רוב מערכות התקשורת של זמן אמתי הן מערכות חשמליות או אלקטרוניות. טלפונים, רדיו וטלוויזיה הם דוגמאות למערכות זמן אמתי.
חלוקה נוספת של מערכות תקשורת נעשית בהתאם לאפשרויות העברת המידע בין המקור והיעד: מערכת תקשורת אשר אינה מאפשרת קשר דו־כיווני בין המקור והיעד נקראת "simplex". מערכת המאפשרת קשר דו־כיווני נקראת "duplex". כדוגמה למערכות תקשורת כאלה נשתמש בטלגרף, שהוא מערכת התקשורת החשמלית הראשונה שהומצאה הפועלת בזמן אמיתי.
מערכת טלגרף הבנויה כמערכת simplex מוצגת בתרשים 4.8. במערכת זו מקור ההודעה הוא המפסק והיעד הוא המנורה. סגירת המפסק גורמת להדלקת המנורה. ליעד אין כל אפשרות לשדר הודעות אל המקור, ולכן איננו יכול לאמת או לברר דבר בקשר להודעה.



תרשים 4.8: מערכת טלגרף מסוג simplex

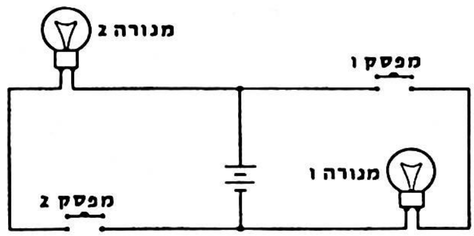
בצורה זו ניתן לשלוח הודעות לפי סימנים מוסכמים (כמו בכתב המורס). אולם כאשר אין המנורה נמצאת בטווח עינו של מפעיל המפסק, לא יכול המפעיל לדעת אם המנורה פועלת, או במלים אחרות, אין באפשרותו לדעת אם נתקבלה ההודעה שנשלחה ממנו.

תרשים 4.9 מציג מערכת פשוטה אחרת של טלגרף, אשר בה ניתן לשלוח ולקבל הודעות בשני קצוות המערכת. מערכת זאת נקראת half duplex.



תרשים 4.9 : מערכת טלגרף מסוג half duplex

במערכת זו סגירת כל אחד מהמפסקים תגרום להדלקת שתי המנורות בעת ובעונה אחת. לכן ניתן להפעיל בזמן מסוים מפסק אחד בלבד, ובכך למנוע בלבול בין ההודעות הנשלחות משני המקורות (המפסקים).
אם יש צורך לשלוח הודעות בו־זמנית משני הקצוות, יש להשתמש במערכת טלגרף הנקראת full duplex. דוגמה למערכת כזאת מוצגת בתרשים 4.10.



תרשים 4.10 : מערכת טלגרף מסוג full duplex

במערכת זאת ניתן להפעיל את המפסקים באותו זמן, כיון שכל אחד מפעיל מנורה אחרת. תנועת ההודעות במערכת היא דו־סטרית וקיימים בה למעשה שני זוגות של מקורות ויעדים.

בדרך־כלל ניתן להשתמש במערכות התקשורת הקיימות להעברת נתונים. המערכת המקובלת היא מערכת הטלפונים הרגילה. ניתן כמובן להשתמש גם במערכות שידור להעברת נתונים.

אמצעי התקשורת בין מערכות ממוחשבות

אמצעי קשר קיימים

קווי טלפון פרטיים

קווי טלפון פרטיים קשורים לרשת הטלפונים הרגילה, שבה נקבע קו אחד כמיוחד למשתמש (קו נקודה לנקודה — נל"ן). בקו הפרטי ניתן להעביר נתונים

ממקור ליעד קבועים, אך לא ניתן להתקשר באמצעות קו זה לקווים אחרים המחוברים לרשת הטלפונים. החיוב עבור קווים אלה הוא לפי תעריף מיוחד, שנקבע בדרך־כלל לפי ק״מ לחודש.

קווי טלקס פרטיים

קווי טלקס פרטיים נמסרים בחכירה על־ידי חברת הטלפונים לשימוש המנויים. לרשת קווי הטלקס ניתן לחבר מסופים לקבלת נתונים ולהעברתם ממקורות שונים ליעדים המחוברים לרשת. בחירת היעד נעשית על־ידי הקשת סימנו או מספרו, בדומה לחיוג ברשת הטלפונים.

אמצעי קשר מתמחים

האמצעים הקיימים כיום לתקשורת יועדו ותוכננו בעיקר עבור ערוצי דיבור. למרות הרצון והמאמצים של חברות הטלפונים להסב עצמן למתן שירותי העברת נתונים, עדיין קיימות מגבלות אשר מונעות ניצול מירבי ונכון של אמצעים אלו להעברת נתונים.

עקב סיבות אלו מוקמות כיום ברחבי העולם חברות המתמחות בהתקנת מערכות תקשורת אך ורק עבור תקשורת המחשבים. המטרה היא לבנות רשת תקשורת אשר תשתלם מבחינה כלכלית ותהיה כמובן, בעלת אחוז שגיאות מינימלי (כלומר, ללא רעשים).

תקשורת נתונים (data communication)

עם התקדמות טכנולוגיית המחשבים נוצר מצב המאפשר למספר רב של משתמשים להיעזר במחשב אחד או במערכת מחשבים. דבר זה ניתן לביצוע רק בעזרת מערכת תקשורת נוחה ואמינה.

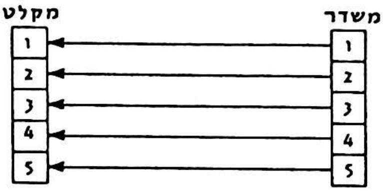
בין המשתמשים הטיפוסיים במערכת תקשורת נמצאים: חברות גדולות בעלות סניפים רבים המרכזות את המלאי או את המנגנון, מדענים במעבדות המשתמשים במערכות מחשב מרוחקות, רשתות בתי־מלון המשתמשות במערכת הזמנת מקומות הקושרת מספר רב של בתי־מלון וכד׳. שימושים אלו לצד מספר רב של אחרים הגדילו את הדרישות למערכות תקשורת.

שיטות ומערכות לתקשורת נתונים

המחשבים הספרתיים פועלים, כזכור, על־ידי הצגה בינרית של הנתונים. גם בהעברת נתונים בין מחשבים לבין מסופים, מיוצגים הנתונים המועברים בצורה בינרית.

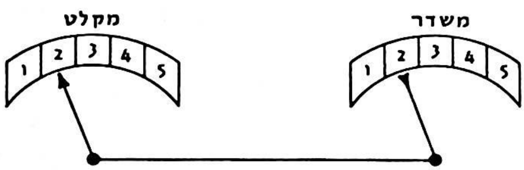
כדי להעביר תווים מוצפנים, יש לסדר את מרכיביהם כך שהצד המקבל יבין אותם. קיימות שתי שיטות עיקריות לבצע זאת: העברה במקביל (parallel transmission) והעברה בטור (serial transmission).

בהעברה במקביל, מועברת כל סיבית של התו במוליך נפרד, כך שכל המרכיבים מועברים בו־זמנית. לדוגמה, אם התו מורכב מחמש סיביות, יש צורך בחמישה מוליכים כדי להעביר את התו כולו. ראה תרשים 4.11.



תרשים 4.11: העברה במקביל

בהעברה בטור, נשלחת כל סיבית של התו בתורה. שיטה זו מאפשרת העברת הנתונים על גבי מוליך אחד בלבד. ראה תרשים 4.12.



תרשים 4.12: העברה בטור

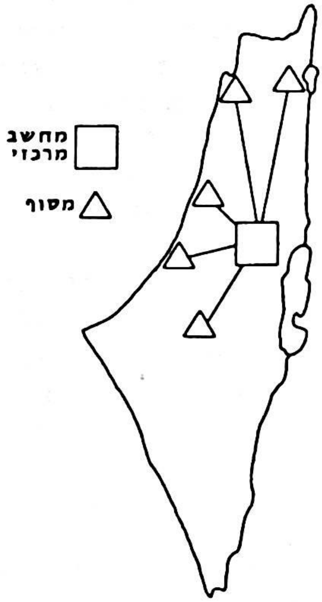
עיבוד רחק (teleprocessing)

"עיבוד רחק" נעשה כאשר קיים מרחק גיאוגרפי בין אתר המחשב לבין מיקום המשתמשים ועל כן הנתונים מועברים ביניהם אלקטרונית. יישומים רחוקים ניתנים לביצוע בין שתי קומות באותו בניין, או בין שני משרדים בשתי יבשות שונות. עיבוד רחק מבוצע באמצעות רשת המעבירה נתונים באופן אלקטרוני. הרשת יכולה לכלול מחשב אחד ומספר מסופים רחוקים, או כמה מחשבים שאליהם קשורים מסופים. הסוג הראשון נקרא רשת מסופים; הסוג השני נקרא רשת מחשבים.

עיבוד רחק מאפשר למפעלים ולארגונים גדולים לרכז את הנתונים של הסניפים המרוחקים במקום אחד מרכזי ולקבל מידע מעודכן על הנעשה בסניפים למרות המרחקים הגדולים. עיבוד רחק מאפשר גישה למאגרי נתונים משותפים, תכניות משותפות וכדומה. ניתן גם לתגבר ביעילות את ניצול המחשבים הכלולים ברשת התקשורת על־ידי הקצאה ותזמון נכונים של העיבודים.

השימוש הנרחב בעיבוד רחק החל למעשה באמצע שנות החמישים, כאשר הוכח שניתן לחבר מכשירים פשוטים (כמו טלפרינטרים) למחשב, להזין את המחשב בנתונים באמצעות מכשירים אלה ולקבל באמצעותם את התוצאות. ההתפתחות החשובה ביותר אשר איפשרה שימושים נרחבים בעיבוד רחק היתה שיטת שיתוף הזמנים (time sharing), אשר איפשרה חיבור מספר מסופים למחשב

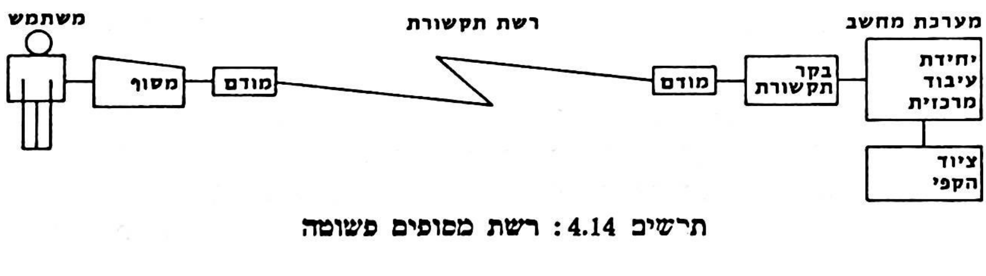
אחד מרכזי, כך, שניתן היה לבצע עיבודים רבים בעת ובעונה אחת. תרשים 4.13 מציג מערכת שבה מחוברים מספר מסופים למחשב אחד מרכזי, דבר המאפשר ביצוע מספר עיבודים מרחוק בעת ובעונה אחת.



תרשים 4.13 : דוגמה למערכת שיתוף זמנים

רשת מסופים (terminal network)

רשת מסופים מורכבת ממחשב מרכזי אחד וממספר מסופים. כל מסוף יכול להיות קשור ברשת תקשורת ישירות למחשב, או מספר מסופים יכולים להוות אשכול המחובר למחשב באמצעות היתקן מתווך. היתקן זה יכול להיות מחשב זעיר, אך זה אינו בלתי־תלוי ואינו בעל עצמה כמו המחשב הראשי. תרשים 4.14 מתאר את הצורה הפשוטה ביותר של רשת מסופים.



תרשים 4.14 : רשת מסופים פשוטה

המשתמש מפעיל מסוף המחובר ברשת תקשורת למחשב. ההמרה מתדירויות קווי הטלפון לאלו של המחשב ובחזרה מבוצעת ע״י היתקן מיוחד הקרוי מודם (modulation — demodulation). קו טלפון רגיל מחבר את המודם לאתר המחשב, שם נמצא מודם אחר המבצע את ההמרה ההפוכה. מודם זה מחובר לבקר תקשורת, שהוא ציוד היקפי מיוחד לתקשורת מרחוק. בקר התקשורת מבוקר ע״י המחשב ומהווה חלק מתצורת המחשב, הכוללת גם היתקני גישה ישירה. לתוכנת המחשב יכולת טיפול בעיבוד רחק בנוסף לעיבודים מקומיים.
ניתן לסווג מסופים ברשת כזו לשני סוגים: **מסופי אצווה מרחוק** (remote batch) **ומסופים אינטראקטיביים**.
מסוף אצווה מרחוק מורכב מהיתקני קלט/פלט מהירים יחסית. המשתמש יכול לשלוח עבודת אצווה למחשב מרוחק ולקבל פלט באמצעות המסוף. לעומת זאת אין הוא יכול להשתמש בתכנית הפועלת בזיכרון המחשב. מסוף אינטראקטיבי כולל בדרך־כלל לוח מקשים ומסך או מכונת כתיבה. המשתמש מבצע באמצעותו ״שיחות״ עם המחשב, כאשר תכנית יישומית מבוצעת בזיכרון. סוגים רבים של יישומים יכולים להישען על עיבוד אינטראקטיבי — הזנת נתונים ישירה, עדכון קבצים בזמן אמתי, הצגת פלט מהירה וכדומה.

רשת מחשבים (computer network)
בניגוד לרשת מסופים, שבה מחשב מרכזי מפקח על כמה מסופים, מורכבת רשת מחשבים ממספר מחשבים, כאשר אף לא אחד מהם נעלה על האחרים. כל מחשב ברשת יכול לפקח על רשת מסופים משלו ויכול לבוא מיצרן שונה. המחשבים ברשת מחוברים ביניהם באמצעות קווי תקשורת, כאשר חלק מהמחשבים משמשים לעיבוד וחלק משמשים לפיקוח על הרשת (communications control systems).

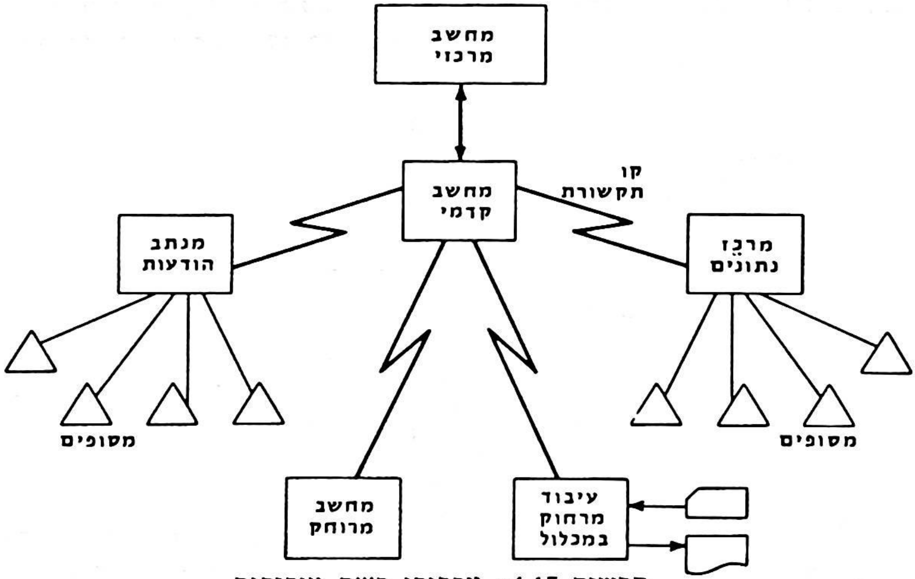
מערכות העיבוד מתחלקות לשני סוגים

א. מחשבים מרכזיים (main-site computers, host computers).
ב. מערכות עיבוד מרוחקות (remote computing systems).

המחשבים המרכזיים ברשת מבצעים את העיבודים העיקריים, מפקחים על מאגרי הנתונים המרכזיים, ובדרך־כלל מפקחים גם על פעילות הרשת. המחשבים המחוברים לרשת יכולים להתחלק ביניהם במשאבים כגון בתכניות, במסדי נתונים ובהיתקני אחסון.

מערכות עיבוד מרוחקות הן בדרך־כלל מחשבים מקומיים אשר להם קשר עם המחשבים המרכזיים של הרשת. הם מבצעים חלק מהעיבודים, אשר בלעדיהם היו מתבצעים על־ידי המחשב המרכזי; כך הם מקלים את הלחץ המוטל על המחשב המרכזי. כמו כן הם מבצעים חלק מהעיבוד הניתן לביצוע במערכת העיבוד המרוחקת, אשר קרוב לוודאי, גם קרובה למקום כניסת הנתונים לרשת.

ברשת מחשבים יכול כל אחד מהמחשבים המרכזיים והמרוחקים לפעול גם כמחשב מקומי. לפיכך יכולה רשת מחשבים לכלול מספר מחשבים מרכזיים, מספר מחשבים מרוחקים, מספר מחשבי פיקוח ומספר רב של מסופים.
תרשים 4.15 מציג רשת טיפוסית הכוללת את רוב מרכיבי הרשת.



תרשים 4.15: מרכיבי רשת טיפוסית

שיתוף משאבים (resource sharing)

סיבות חשובות להתעניינות ההולכת וגוברת ברשתות מחשבים הן היתרון הכלכלי והנוחות המושגים בעזרת השיתוף במשאבים. על-ידי חיבור מספר מחשבים ניתן ליצור כוח חישוב משותף גדול, ניתן להשתתף במסדי נתונים גדולים ובתוכנה מיוחדת. כל המשאבים הנמצאים באתרים השונים של הרשת עומדים לרשות כל אחד מהמשתמשים המחוברים לרשת.

בשיתוף משאבים נכללים:

שיתוף בציוד (device sharing) — היכולת של מחשב הקשור לרשת להתחבר לציוד מרוחק, כאילו נעשה החיבור לציוד מקומי.

שיתוף קבצים (file sharing) — היכולת לקרוא, לכתוב או לעדכן קבצי נתונים הנמצאים במקום מרוחק, כאילו היו מקומיים.

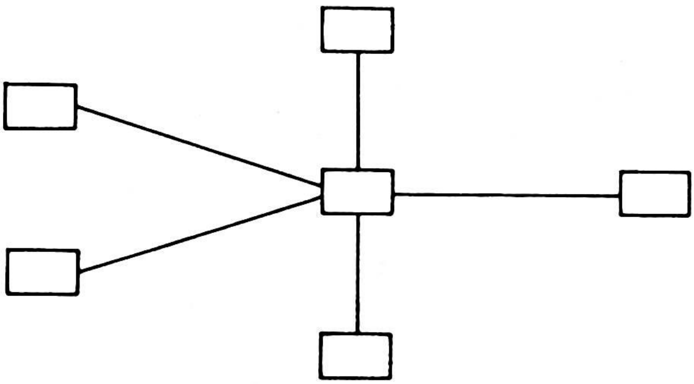
שיתוף בתכניות (program sharing) — היכולת של מחשב אחד להשתמש בתכניות הנמצאות במקומות מרוחקים.

שיתוף בנתונים לתכניות (program data sharing) — היכולת להעביר נתונים בין תכניות מחשב המופעלות במקומות מרוחקים, כך שעבודות גדולות מפוצלות למספר עבודות קטנות ומתבצעות במספר מקומות.

סוגי רשתות

מבנה מרוכז או מבנה כוכב (centralized or star structure)

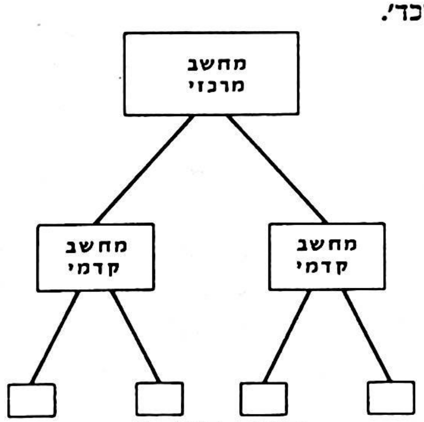
דוגמה למבנה הכוכב מוצגת בתרשים 4.16. בשיטה זו מתקשרים כל המחשבים ברשת ישירות למחשב מרכזי (כוכב) באמצעות קווי תקשורת נפרדים. כאשר יש להעביר הודעות בין המחשבים, משמש המחשב המרכזי כמנתב ההודעות. מבנה זה מאפשר פיקוח פשוט על הרשת.



תרשים 4.16: מבנה כוכב

מבנה דמוי עץ (tree structure)

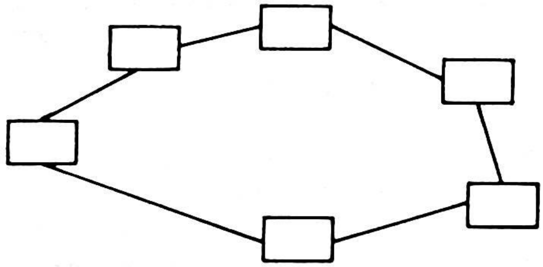
מבנה זה מקובל בעיקר בתעשייה, מאחר שבדרך־כלל יש צורך בפיקוח ובניהול של יישומים תעשייתיים מסוימים בזמן אמתי (תרשים 4.17). במערכות במבנה דמוי עץ ניתן לפקח על תהליכים, לכוונם ולדווח על מצבם: מערכות גישוש מספקות למערכת הפיקוח נתונים בזמן אמתי, המחשבים ברמת הפיקוח מאמתים את התהליכים ובמקביל מעבירים נתונים למחשב המרכזי, המעבד מידע הקשור בתכנון, בקרה, ניהול וכד׳.



תרשים 4.17: מבנה עץ

מבנה טבעת (ring structure)

ארגונים רבים תכננו את רשת המחשבים שלהם בצורת מבנה הטבעת. מבנה זה אינו מאפשר למספר רב של מחשבים מרוחקים המחוברים לרשת להתקשר ישירות ביניהם. בשיטה זו מועברים הנתונים סביב המחשבים (תרשים 4.18).

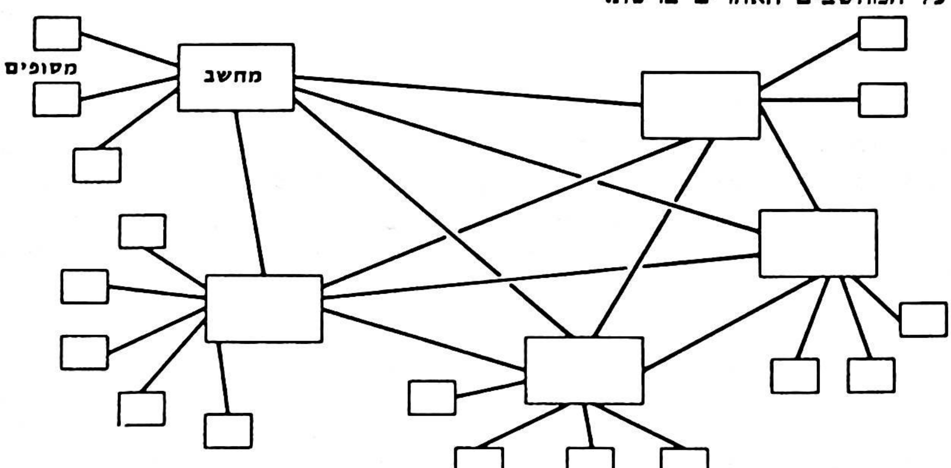


תרשים 4.18: מבנה טבעת

מבנה כזה כדאי מבחינה כלכלית כאשר המחשבים נמצאים קרוב זה לזה. כאשר המחשבים מרוחקים בצורה ניכרת יהיה מבנה זה יקר מדי עקב עלות העברת הנתונים בקווי התקשורת; במקרה כזה יהיה עדיף המבנה המפולג (תרשים 4.19).

מבנה מפולג (distributed structure)

תרשים 4.19 מראה רשת מפולגת הכוללת מספר מחשבי פיקוח או ניתוב, המפקחים על תנועת הנתונים ברשת.
מבנה כזה עשוי להיות חסכוני, אם ליד כל רשת מסופים ימצא מחשב שאליו התחברו המסופים הקרובים אליו גיאוגרפית. אם רשת מפולגת מתוכננת נכון, היא יכולה להיות אמינה ביותר, מאחר שתקלות במחשב מסוים אינן משפיעות על המחשבים האחרים ברשת.



תרשים 4.19: רשת בעלת מבנה מפולג

שיטות עיבוד של רשתות מחשבים ומסופים

ניתוב הודעות (message switching and routing)

ניתוב הודעות היא שיטת הטיפול בהעברת נתונים ברשתות תקשורת. הנתונים מועברים לנקודת ביניים (מחשב מנתב), מאוחסנים לתקופת זמן, בדרך־כלל קצרה מאוד, ולאחר מכן מועברים אל היעד.

קיימות שלוש שיטות בסיסיות בניתוב הנועת תקשורת מהמקור ליעד ובחזרה:

ניתוב מעגלים (circuit switching)

ברשת עם מתוג מעגלים, קיימת מרכזיה שתפקידה ליצור קשר ישיר בין מסוף או מחשב למשנהו. לאחר יצירת הקשר, המסופים מעבירים את המידע ביניהם. כאשר מסתיימת העברת ההודעה, מנתקת המרכזיה את המעגל ומחזירה את המערכת למצב נכונות לקראת ההתקשרות הבאה. רשת הטלפונים הרגילה פועלת בצורה זו.

ניתוב הודעות (message switching)

בשיטה זאת, כל הודעה הנשלחת לרשת מנותבת כיחידה אחת ליעד. הודעות נוספות בין אותו מקור ליעדו יכולות להישלח בנתיב אחר לגמרי. כך שלמעשה הקשר בין מקור מסוים ליעדו הוא לא קבוע כמו בשיטה הקודמת, אלא נקבע בהתאם למצב הרשת בזמן שבו נוצר הצורך לשלוח הודעה. כל הודעה נשלחת עם כתובת היעד. כיון שלעתים יכול להיווצר מצב שאחת מתחנות היעד שאליה נשלחה ההודעה עסוקה, מאוחסנת ההודעה בתחנת ביניים עד אשר תתפנה תחנת היעד. לפיכך מכונה שיטה זו גם בשם "אחסן ושלח" ("store and forward" system).

ניתוב צרורות (packet switching)

שיטה זאת דומה במהותה לשיטת ניתוב הודעות, אלא שההודעות הארוכות מפוצלות במקור לחלקים בעלי אורך קבוע (הנקראים צרורות), ואלו נשלחים, כל צרור בנפרד, בדרך היעילה ביותר. כן מאפשרת שיטה זאת לשלוח הודעות ארוכות בזמן קצר, כיון שההודעה מפוצלת לחלקים קטנים, הנשלחים בו־זמנית בנתיבים שונים לאורך הרשת.

פונקציות הרשת (network functions)

רשתות מחשבים ומסופים מחוברות באמצעות קווי תקשורת.
רשת יכולה להיות בנויה ממסוף המחובר למחשב ועד לרשת מורכבת הכוללת מסופים, מרכזי נתונים, מחשבים מרוחקים ומחשבים מרכזיים. כל מחשב במערכת

כזאת נבחר לפי יכולתו לבצע את תפקידו במקומו הפיסי וברשת. המחשבים יכולים להיות בעלי מהירויות פעולה שונות, שפות תכנות שונות ומאפיינים שונים אחרים. החשוב מכול, שיהיו יעילים בביצוע הפונקציות שנועדו להם.

עיבוד מרכזי (host processing)

מערכת העיבוד המרכזית יכולה להיות מחשב מרכזי גדול במערכת שיתוף זמנים טיפוסית. היא יכולה לקלוט ולעבד קבוצה כללית של דרישות ממספר רב של היתקנים מקומיים או מרוחקים, כולל מחשבים ומסופים. המחשב המרכזי הוא בעל זיכרון גדול ומכלול רחב של ציוד היקפי, הכולל בין השאר תקליטים מגנטיים, המאפשרים אחסון גדול של נתונים, ומשמשים כמסד נתונים לכל הרשת. מערכת התוכנה של המחשב המרכזי כוללת מערכת הפעלה מתוחכמת המפקחת על פעולות הרשת, והנתמכת על־ידי מערכת תוכנה לתקשורת, שתוכננה במיוחד עבור הפעלת הרשת.

בנוסף, מבצע המחשב המרכזי חלק מרבי מהעיבודים הנדרשים על־ידי המשתמשים ברשת.

עיבוד קדמי בתקשורת (front-end communications processing)

שיטה זו מאפשרת שחרור המחשב המרכזי מטיפול במערכת התקשורת. הפיקוח על התקשורת נעשה על־ידי מחשב "קדמי" שזה תפקידו העיקרי. בתפקיד זה משמשים מחשבים זעירים, המותקנים ברשת בין המסופים למחשב המרכזי. שיטה זו עדיפה מבחינה כלכלית, כיון שהמחשב המרכזי, היקר בדרך־כלל בהרבה ממחשב זעיר, משוחרר מטיפול בבעיות התקשורת.

ריכוז מרוחק (remote concentration)

בשיטה זו מחוברים מספר מסופים בעלי מהירות העברת נתונים קטנה יחסית למרכז נתונים, הקשור למחשב בקו תקשורת מהיר יותר. מרכז הנתונים (data concentrator) אוסף נתונים ממספר מסופים אטיים ומעבירם הלאה במהירות גבוהה יותר. לביצוע תפקיד זה ניתן להשתמש גם כאן במחשבים זעירים. ריכוז מרוחק מאפשר הקטנת עלות הקווים ברשת התקשורת.

הזנת עבודות מרחוק (Remote Job Entry — RJE)

קיימים ברשת מסופי אצווה מרחוק המאפשרים עיבוד באצווה מרחוק (remote batch processing). בשיטה זאת מועברת עבודה (תכנית ונתונים) למחשב המרכזי ומצטרפת לעבודות הממתינות לתורן לעיבוד. לאחר ביצוע העבודה נשלחות התוצאות חזרה לנקודה שממנה הוזנה לרשת.

מערכת מחשב מרוחקת (remote computer system)
שלא כשיטת העיבוד באצווה מרחוק, מסוגל המחשב המרוחק לבצע חלק מהעיבודים בעצמו. רק באותם מקרים שלא ניתן לבצע בהם את העיבוד במחשב המרוחק, נשלחת העבודה לביצוע למחשב המרכזי.

חלק שלישי

יישום מערכות מידע
ממוחשבות בארגון

פרק 5.

יישום מערכות מידע ממוחשבות

המפתח להצלחת תפקודן של מערכות המידע בארגון היא מערכת תכנון כוללנית ויעילה.

התפקוד במערכות המידע זהה לתפקידים ארגוניים אחרים (כמו מימון, שיווק, כוח־אדם, ייצור, חשבונאות) בכך שהמערכת מורכבת ממנהלים ברמות שונות החייבים לתכנן כיצד לנצל ביעילות את המשאבים המוגבלים העומדים לרשותם לצורך ביצוע פעילויותיהם.

בפרק 1 סווגו מנהלים לפי — הדרג האסטרטגי, הדרג הטקטי והדרג התפעולי. דרגי הניהול בפונקצית מערכות המידע ערוכים מלמעלה למטה. בראש המערכת עומד ראש יחידת מערכות המידע, לאחריו — מנהלי תכנות, תיכון מערכות ותפעול, ולבסוף מנהלי פרוייקטים, תכניתנים ראשיים, מפעילים ראשיים וכד' (תיאור בעלי תפקידים אלו נמצא בפרק 6).

מערכת התכנון במערכות המידע חייבת להיות חלק אינטגרלי מהתכנון הארגוני הכולל. חשוב שתכניות מערכות מידע ישתלבו במערכת התכנון של הארגון. התכניות חייבות להיות מותאמות לאפיונים הייחודיים של כל ארגון. תכנון מערכות מידע חייב להתאים לתנאים השוררים בכל ארגון ועליו להשתנות כאשר התנאים משתנים.

תהליך שלם של תכנון מערכות מידע מתחיל עם קביעת תכנית אסטרטגית, אשר מספקת מסגרת לפיתוח תכניות ברמות הנמוכות. כל שלב בתהליך התכנון מהווה פירוט של שלבים קודמים, כאשר התהליך כולו מסתיים בתכנון יישומים תפעוליים בדידים. ניתן אפוא לדבר על היררכיה של תכניות מערכות מידע.

תכנון אסטרטגי — זהו התהליך שבמהלכו נקבעים המטרות של פונקצית

פרק 5.

יישום מערכות מידע ממוחשבות

המפתח להצלחת תפקודן של מערכות המידע בארגון היא מערכת תכנון כוללנית ויעילה.
התפקוד במערכות המידע זהה לתפקידים ארגוניים אחרים (כמו מימון, שיווק, כוח־אדם, ייצור, חשבונאות) בכך שהמערכת מורכבת ממנהלים ברמות שונות החייבים לתכנן כיצד לנצל ביעילות את המשאבים המוגבלים העומדים לרשותם לצורך ביצוע פעילויותיהם.
בפרק 1 סווגו מנהלים לפי — הדרג האסטרטגי, הדרג הטקטי והדרג התפעולי. דרגי הניהול בפונקצית מערכות המידע ערוכים מלמעלה למטה. בראש המערכת עומד ראש יחידת מערכות המידע, לאחריו — מנהלי תכנות, תיכון מערכות ותפעול, ולבסוף מנהלי פרוייקטים, תכניתנים ראשיים, מפעילים ראשיים וכד' (תיאור בעלי תפקידים אלו נמצא בפרק 6).
מערכת התכנון במערכות המידע חייבת להיות חלק אינטגרלי מהתכנון הארגוני הכולל. חשוב שתכניות מערכות מידע ישתלבו במערכת התכנון של הארגון. התכניות חייבות להיות מותאמות לאפיונים הייחודיים של כל ארגון. תכנון מערכות מידע חייב להתאים לתנאים השוררים בכל ארגון ועליו להשתנות כאשר התנאים משתנים.
תהליך שלם של תכנון מערכות מידע מתחיל עם קביעת תכנית אסטרטגית, אשר מספקת מסגרת לפיתוח תכניות ברמות הנמוכות. כל שלב בתהליך התכנון מהווה פירוט של שלבים קודמים, כאשר התהליך כולו מסתיים בתכנון יישומים תפעוליים בדידים. ניתן אפוא לדבר על הירככיה של תכניות מערכות מידע.
תכנון אסטרטגי — זהו התהליך שבמהלכו נקבעים המטרות של פונקצית

מערכות המידע בארגון, המשאבים הדרושים להשגת המטרות והעקרונות שינחו את הרכישה, השימוש והשחרור של המשאבים.
תכנון **לטווח ארוך** — עוסק בקביעת צורכי המידע העתידיים של הארגון ובמבנה יחידת מערכות המידע בעתיד. התכנון מקיף חמש עד עשר שנים והוא נושא אופי מושגי.

תכנון **לטווח בינתי** — עוסק בקביעת צורכי המידע הנוכחיים של הארגון, החזויים שנתיים עד חמש שנים קדימה. התוצר של תכנון זה, הנושא אופי ניהולי, מהווה את תכנית האב לפיתוח מערכות מידע בארגון. תכנית האב קובעת את הקדימויות לפיתוח מערכות או יישומים בדידים.

תכנון **לטווח קצר** — זהה לתכנית השנתית של פונקצית מערכות המידע ונושא אופי תפעולי. הוא כרוך בלוחות זמנים לפיתוח יישומים בדידים, הכנת תקציב, תזמון כוח־אדם וקביעת יעדים לביצוע.

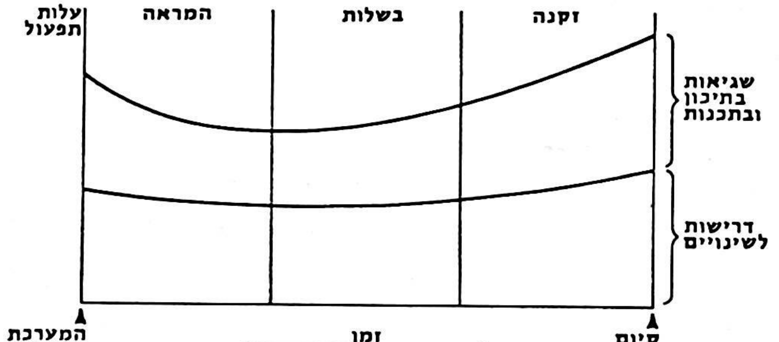
מחזור החיים של מערכת מידע ממוחשבת

"מחזור החיים" הוא למעשה תהליך הפיתוח והיישום של מערכת מידע ממוחשבת. מחזור החיים מורכב משלבי הפיתוח שדרכם עוברת כל מערכת מידע: החל מדרישה ראשונית למידע (צורך או הזדמנות שצצו בארגון), דרך התיכון, הפיתוח והתפעול של המערכת עד לסיום חיי המערכת, ומכאן — להתחלת מחזור חיים של מערכת חדשה. הגישה נושאת אוריינטציה ניהולית; היא מדגישה את האופי הלוגי של **מה** צריך לעשות ולא את המפרטים הטכניים הכרוכים **באיך** לבצע את השלבים השונים של מחזור החיים.
קיימים תיאורים רבים של שלבי מחזור החיים במערכת מידע, אולם הם אינם משקפים גישות שונות לנושא. כולם מתחילים בקביעה של מה על המערכת לעשות, עוברים דרך שלב התיכון, שלב התקנה ושלב התפעול ונגמרים בנקודה שבה המערכת הגיעה להתיישנות.
מחזור החיים של מערכת מידע ממוחשבת הוא תהליך המגיע אל סופו בתום ארבע עד עשר שנים. הוא מתחיל ונגמר בהכרה שצורכי המידע אינם מסופקים ביעילות ע"י מערכת המידע הקיימת. כל המערכות פג זמנן בנקודה מסוימת, שממנה חייב להתחיל מחזור חיים חדש.
בפרק 1 נדונו המאפיינים השונים של כל מערכת — מטרה, סביבה, גבולות, אילוצים, קלט, פלט, רכיבי המערכת וקשרי הגומלין ביניהם. כל המאפיינים משתחקים במידה רבה או מועטה במהלך השוטף של חיי המערכת, והתוצאה — יעילות המערכת יורדת, ויש צורך להשקיע בשינויים; עם הזמן, השינויים מותקנים האחד על גבי השני. לבסוף, עלות השינויים אינה מצדיקה את התועלת שבהתקנתם והארגון מחליט להחליף את המערכת הקיימת.

הגורמים לשינויים המשפיעים על מאפייני המערכת יכולים להיות טכנולוגיים, כלכליים, ארגוניים ואנושיים.

למשל:
— גידול בנפחי התנועות (הארגון גדל עם הזמן)
— התפתחויות טכנולוגיות (קווי תקשורת אמינים וזולים)
— שינויים במערכות קשורות (למערכת ממוחשבת אחרת נוספו צגים גרפיים)
— שינויים בסביבה (טבלאות הניכויים למס-הכנסה שונו)
— שינויים בציפיות המשתמשים ובדרישותיהם (הצרכן העיקרי של המערכת דורש דוחות יומיים במקום שבועיים)

חלק מהגורמים הללו יכול להביא לייזום שינויים חופפים. טיפוסי, למשל, שהופעת טכנולוגיות חדשות תגרום לדרישות חדשות מצד המשתמש. כאשר השינויים קטנים יחסית, הם נכללים בפעילות האחזקה השוטפת של מערכת מידע, וגם כאשר הם רציניים יותר הם עדיין עשויים להיכלל במסגרת השיפורים במערכת. אבל כאשר השינוי הוא מהותי, או כאשר מצטברת כמות גדולה של שינויים קטנים, תתעורר דרישה להחלפת המערכת הקיימת. תרשים 5.1 מציג את הצטברות האירועים הללו במהלך חיי המערכת. בתקופה הראשונה של פעילות המערכת, שגיאות בתיכון ובתכנות ודרישות לשינויים נוטים לרדת ולהקטין את עלות התפעול. בתקופת הבשלות של המערכת, כמות השגיאות והדרישות היא יציבה יחסית. בתקופת הזקנה של המערכת, עלות התפעול עולה עד לנקודה שממנה מוצדק המעבר למערכת חדשה.



תרשים 5.1: התפתחות טיפוסית של עלות תפעול מערכת מידע

מחזור החיים של מערכת מידע ממוחשבת מקיף את הנושאים האלה:
א. מאפייני הסביבה שבה מתבצע תהליך הפיתוח וההפעלה
ב. היבטים ניהוליים של תהליך הפיתוח והיישום

ג. הפעילויות בשלבי הפיתוח והתפעול ומבצעיהן
ד. ארגון תהליך הפיתוח.

מערכת מידע ממוחשבת מורכבת ממספר רב של רכיבים המקיימים ביניהם יחסי גומלין. בתהליך הפיתוח והתפעול משתתפים גורמים רבים ושונים מכל חלקי הארגון שהם בעלי רקע מקצועי, ניהולי ומעמדי שונה. מצב זה, המבטא את המורכבות וההיקף של הפיתוח והתפעול של מערכות מידע, מכתיב את הצורך שהתהליך יוכנס למסגרת מוגדרת ובת־שליטה. כך ניתן לחלק את מערכת המידע לתת־מערכות תוך הגדרת הממשקים ביניהן, ולטפל ברכיבים בני־שליטה, בני־פיתוח ובני־בקרה תוך ידיעה, שבסופו של התהליך יתחברו הרכיבים למערכת הרמונית שתשיג את מטרותיה ביעילות מרבית.

רעיון מחזור החיים אינו מוגבל למערכות מידע בלבד. תפיסה דומה מיושמת, לדוגמה, בפיתוחו של מוצר חדש. ההבדל בין מערכות מידע למוצר כלשהו הוא בכך, שהמוצר הוא עצם פיסי־מוחשי ואילו מערכות המידע הן עצם מופשט. הרעיון המרכזי הטמון ביסוד מחזור החיים הוא, שפיתוחה ותפעולה של כל מערכת חייבים לעבור דרך תהליך שיטתי ולוגי המורכב משלבים מוגדרים שעליהם לא ניתן לפסוח ושאת ביצועם אין לשנות. בנייתה של מערכת מידע ממוחשבת דומה לבניין בית, שהוא מוצר חד־פעמי ברוב המקרים: אין דומה בית אחד למשנהו, אולם תהליך הבנייה זהה.

להלן חלק מהבעיות שבהן מתלבטים ארגונים המפתחים מערכות מידע ממוחשבות:

— השקעות גדולות מהמתוכנן הן בפיתוח והן ביישום
— לוח זמנים לפיתוח ארוך מדי
— במשך זמן הפיתוח חלים שינויים ארגוניים וסביבתיים המשפיעים על המערכת שבפיתוח
— אין המפתחים עומדים בלוח הזמנים המתוכנן
— דרישות עיקריות למידע אינן נכללות במערכות שמפתחים
— התקנת המערכות בארגון כרוכה במאמצים רבים ובתקלות בלתי־צפויות
— הפלט, גם לאחר תום הפיתוח, אינו אמין
— התועלת המתוכננת אינה מושגת.

להלן הסיבות השכיחות הגורמות לבעיות אלו:

— חוסר מעורבות של הנהלת הארגון והמשתמשים העתידיים בתהליך ההגדרה והפיתוח.
— אין מקדישים די זמן לשלב ההגדרה. תוצאותיו הן מפרטים שטחיים ובלתי שלמים.
— היקף המערכות רחב מדי.
— אי־קיום מסגרת שיטתית שלמה וברורה לתכנון ולבקרה של תהליך הפיתוח.

— בעת הפיתוח מצמצמים את ההיקף הרחב, ולפיכך תועלת המערכת אינה זו המובטחת.

הצעדים שיש לנקוט כדי למנוע בעיות אלו:

— מיבנות תהליך הפיתוח של מערכות מידע ממוחשבות
— קביעת נקודות בקרה והחלטה מוגדרות, ונוהל ברור לפעילות בנקודות אלו
— קביעת תחומי אחריות וסמכות ברורים בין הנהלת הארגון, הצוות המפתח והמשתמשים העתידיים.

מאפייני הסביבה שבה מתבצע תהליך הפיתוח וההפעלה של מערכות מידע ממוחשבות

א. ההחלטה לפתח מערכת מידע ממוחשבת מתקבלת בשלב מוקדם יחסית של תהליך הפיתוח. מקבלי ההחלטה אינם יכולים בשלב זה לדעת מה מהותה של המערכת שהחליטו להשקיע בה משאבים רבים יחסית.

ב. זמן הפיתוח של המערכות ארוך. זאת, משום שהמערכות של היום מורכבות ובעלות היקף רחב (הן משום שזהו הצורך והן משום שניתן לפתח מערכות כאלו).

ג. התפתחויות מהירות ומשמעיות בשטח החומרה והתוכנה. השפעת התפתחויות אלה על כדאיות או אי־כדאיות מיחשובן של מערכות מידע הוא רב.

ד. היקף המשאבים הדרושים לפיתוח וליישום של מערכות המידע הממוחשבות הולך וגדל עם הזמן.

ה. ההיצע של כוח־האדם המקצועי האמור לעסוק בפיתוח וביישום של מערכות המידע הממוחשבות אינו מדביק את הביקוש, הן מבחינת איכותו והן מבחינת כמותו.

ו. קיימים קשיי תקשורת עקרוניים בין העוסקים בפיתוח לבין המשתמשים במערכות (הלקוחות). עובדה זו באה לידי ביטוי הן בשלב איתור צורכי המידע, הן במהלך תהליך הפיתוח והן בעת היישום של המערכות.

היבטים ניהוליים של תהליך הפיתוח והיישום של מערכות מידע ממוחשבות

א. ההשקעה בפיתוח וביישום מערכות מידע ממוחשבות היא השקעה לכל דבר ויש לטפל בה כפי שמטפלים בכל השקעת הון, כלומר יש להצדיק את ההשקעה. גורמים אלה צריכים אפוא להילקח בחשבון:

— הערך הנוכחי של המערכת לאור ההשקעות בה בהווה ובעתיד (הוצאות הפיתוח ועלות היישום)
— אורך החיים המתוכנן של המערכת
— מחיר הגרט של המערכת הישנה המוחלפת במערכת החדשה שמפתחים

— בסביבה של משאבים מוגבלים, יש לבדוק אם המערכת שעומדים ליישם היא בעלת הערך הגבוה ביותר מבין המערכות שניתן לפתח.

ב. לכל מערכת מידע חייבים להיות מדדים כמותיים שיאפשרו לאמוד את מידת הצלחתה של המערכת, כך שניתן יהיה לדעת באיזו מידה השיגה את מטרותיה. רצוי שהמערכת עצמה תפיק מידע שיאפשר את האמדת הצלחתה.

ג. המערכת שאותה מיישמים נועדה לשרת את המשתמש. פירושה של הקביעה שדרושה מעורבותו הפעילה של המשתמש בתהליך הפיתוח של מערכת המידע. המפרטים של המערכת והשינויים במפרטים אלה חייבים לקבל את אישורו של המשתמש לפני המשך העבודה. המצב הרצוי הוא שיתופו הפעיל של המשתמש בבניית המפרטים. המשתמש חייב ליטול על עצמו את האחריות להתקנתה של המערכת בארגונו ולאשר את נוהלי השימוש במערכת לפני הפצתם. המצב הרצוי במקרה זה הוא שיתופו של המשתמש בכתיבת הנהלים.

ד. מאחר ותהליך הפיתוח של מערכת מידע ממוחשבת ארוך ומורכב, יש לחלקו לקטעים לוגים בני־בקרה. בסופו של כל קטע ניקבעות נקודות בקרה (אבני־דרך) שבהן מתבצעת בקרה על העבודה שנעשתה ועל המשאבים שנצרכו עד לנקודת הבקרה. כמו כן מופעלת בקרה על תכנון העבודה ועל המשאבים שיש לבצע עד לסיום העבודה.

תהליך הפיתוח וההפעלה של מערכות מידע מתחלק לארבעה שלבים עיקריים (המתחלקים לתת־שלבים):

1. שלב ההגדרה
2. שלב הבנייה
3. שלב ההתקנה
4. שלב התפעול והאחזקה.

שלושת השלבים הראשונים מהווים את תהליך הפיתוח. השלב הרביעי מהווה את תהליך התפעול. בבקרת תהליך הפיתוח טמונים מספר קשיים: מבקר מן החוץ אינו "חי" את המערכת טרם תפעל; המפתחים מגינים ב"חום" על עבודתם; בדרך־כלל עומד תהליך הפיתוח בלחץ זמן; בדרך־כלל קיימים גורמים אובייקטיביים המפריעים לתכנון המקורי ולתהליך הבקרה; קיימת נטייה להקצות לעבודת הבקרה עובדים בדרג נמוך, שאינם מסוגלים לעמוד בלחץ מפתחי המערכת.

ה. אין להסס מלהפסיק פיתוחה של מערכת. השלב שבו רצוי לקבל החלטה כזו הוא שלב ההגדרה או במועד המוקדם האפשרי ביותר, משמסתבר שהמערכת אינה ישימה מסיבה כלשהי. זאת, משום שהמשאבים המבוזבזים בגלל ההפסקה אינם רבים עדיין. לשון אחר, בפיתוחה של מערכת מידע ממוחשבת כרוכים סיכונים, שהעיקריים הם:

— המערכת אינה זו הנכונה (אינה עונה על צורכי הארגון)

— המערכת אינה ישימה מהבחינה הטכנית
— המערכת אינה ישימה מהבחינה התפעולית

שלב ההגדרה הוא השלב שבו יש להסיר במידה מרבית סיכונים אלה. לפיכך יש להקצות לשלב ההגדרה, שהוא הפחות ממוסד ושבו מומנט היצירתיות בא לידי ביטוי, את אנשי המקצוע הטובים ביותר. אין צורך ללחוץ על קיצורו של שלב זה, אלא להקצות לו משך זמן שיאפשר הסרת הסיכונים, מה עוד ששלב זה הוא יחסית הזול בשלבי הפיתוח של המערכת. מודגש אפוא לא לחסוך בזמן ובהוצאות בשלב ההגדרה, כדי להקטין את רמת הסיכון בהמשך הפיתוח.

שלב הגדרה גרוע עלול לגרום לבעיות בשלבים הבאים (כגון שינויים במפרטי התיכנות בשלב התיכנות). ככלל, רצוי לבדוק את ישימות המערכת בתום כל שלב עיקרי, לאחר הפירוט הנוסף שהושג בשלב זה.

ו. המעורבות האקטיבית של הנהלת הארגון בתהליך הפיתוח חייבת לבוא לידי ביטוי בעיקר בשלב ההגדרה, עקב חשיבותו של שלב זה. ביצוע נכון של שלב ההגדרה מבטיח במידה רבה את הצלחתם של השלבים הבאים אחריו. נוסף על שלב ההגדרה, נדרשת מעורבות הנהלת הארגון בכל שלבי הפיתוח והיישום. עקרונית, אין להשאיר את ניהוג פעילות הפיתוח והבקרה על המערכות בידי אנשי המקצוע בלבד. גם הנהלת הארגון וגם המשתמשים חייבים להיות אחראים לפעילות זו.

ז. מערכות מידע ממוחשבות הן מוצר עתיר עבודה. בשנות החמישים היוו ההוצאות לציוד 75% מההשקעה בפיתוח מערכת לעומת שיעור של 25% לכוח־אדם, ואילו היום ההוצאות לציוד הן בשיעור של 25% לעומת 75% של הוצאות לכוח־אדם מכלל ההשקעה. מכאן החשיבות שיש לייחס למינוי האחראים לפיתוח ולאיוש הצוות שיעסוק בפיתוח.

ח. תהליך הפיתוח של מערכות מידע ממוחשבות הוא תהליך איטרטיבי שתחילתו בתמונה הכללית והמשכו בירידה לפרטים. טבעי הדבר, שעם הירידה לפרטים מתברר לעתים שקיימת אי־התאמה בין הפרטים לבין המסגרת שנקבעה בשלב מוקדם, ואז קיים הצורך לחזור לשלב הקודם וליישר את ההדורין.

יש לשאוף שמספר האיטרציות (חזרות לשלב מוקדם) יהיה קטן ככל האפשר, באשר כל איטרציה היא מקור לשינויים ולעיכובים. יש לבדוק היטב את השפעת השינויים הקטנים יחסית על תהליך הפיתוח ולהיפך, וכמו־כן יש לוודא שהאיטרציות לא יתחילו בכל פעם מנקודת אפס. ברור שטיב העבודה בשלבים המוקדמים משפיע באופן מכריע על מספר האיטרציות.

ט. בעת הפיתוח של מערכות מידע ממוחשבות יש לבדוק מספר חלופות ישימות ולא להסתפק בפתרון אחד. בדרך־כלל בוחן הצוות המפתח מספר חלופות ישימות, משווה ביניהן וממליץ על הטובה ביותר. במקרים קריטיים ניתן

לתת את אותה המערכת לבחינה לשני צוותים. בחינתן של מספר חלופות תתבצע בשלב ההגדרה.

י. יש לחלק את מערכת המידע שמפתחים לחוליות (מודולים). עקרון זה חל במיוחד על מערכות מידע מורכבות ובעלות היקף רחב. החלוקה לחוליות (או תת־מערכות) מאפשרת טיפול במרכיבים בני־שליטה ומאפשרת פיתוח של צעד אחר צעד, דבר המקטין את הסיכון שבפיתוח. עם זאת חייב האחראי על הפיתוח לראות כל הזמן את המערכת בשלמותה, ולדאוג לכך שבסופו של דבר יתחברו תת־המערכות למערכת הרמונית אחת שתשיג את מטרותיה ביעילות מרבית. יש להימנע מבניית חוליות בעלות היקף רחב.

יא. מאחר שמערכות מידע הן מוצר מופשט, הדרך היחידה שבאמצעותה ניתן לחזות בתוצאות ולבקרן היא התיעוד (דוקומנטציה). לתיעוד התכונות האלה:

— הכתיבה וניסוח בהיר וחד־משמעי משמשים כלי לאיתור חולשות וחללים בתכנון מזה והוכחה לשלמות התכנון מזה
— מאחר שבתהליך הפיתוח מעורב יחסית מספר רב של אנשים, יש צורך באמצעי תקשורת. התיעוד משמש כלי תקשורת מעולה
— התיעוד הוא כלי שבאמצעותו ניתן לנהל את עבודת הפיתוח, לבקר את הנעשה ולתכנן את שייעשה.
— התיעוד משמש גיבוי למקרה שתכניותיה של מערכת הנמצאת בתפעול תהרסנה.
— התיעוד מאפשר תפעול המערכת. מפתחי המערכת אינם בהכרח האנשים אשר יתחזקו ויפעילו אותה. ללא תיעוד לא יוכלו אלה שלא פיתחו את המערכת לתחזקה ולתפעלה
— התיעוד הוא אמצעי הדרכה חיוני.

קיימים סוגים רבים של תיעוד. לכל סוג ייעודו שלו, מבנהו ושפתו. לכל שלב בתהליך הפיתוח תיעוד משלו, שהוא מוצרו של אותו שלב.

יב. התיעוד אינו כלי בקרה דפנסיבי או כלי תקשורת בלבד. תקני תיעוד משמשים כלי למיבנות (structuring) של תהליך הפיתוח. תקנים אלה משמשים כרשימת תיוג (check-list) לעוסקים בפיתוח ומאפשרים להם לבצע את עבודתם במסגרת מוגדרת, אחידה ומכוונת. תהליך הפיתוח המבוסס על תקני תיעוד מפורטים ושלמים הוא תהליך מוגדר מראש, המונע את הצורך לפתח לכל מערכת את מסגרת עבודתה.

יג. בתהליך הפיתוח משתתפים אנשים רבים בעלי רקע מקצועי ומעמדי שונה: הנהלת הארגון, המשתמשים, מרכזי החישובים, מנתחי מערכות, תכניתנים, כותבי נהלים ואחרים. עובדה זו מכתיבה את הצורך למצוא מסגרת ארגונית שתאפשר את עבודתם המשותפת.

יד. תכנון תהליך הפיתוח של מערכת מידע ממוחשבת הנו פעולה מורכבת. אולם ללא תכנון מוקדם אין לגשת לביצוע. אם לא ניתן לתכנן את התהליך, סימן שאין אפשרות לבצעו. אין זה אומר, כמובן, שהתכנון אינו בר־

שינוי לאורך זמן. ראוי לציין כי תכנון עבודה שהיסוד היצירתי בה הוא בעל משקל, אינו פעולה קלה לביצוע. אבל חלוקתו של תהליך הפיתוח לשלבים, תכנון לוח־זמנים והקצאת משאבים לכל שלב ושלב מאפשרים תכנון שהוא טוב בהרבה מתכנון לגבי התהליך כשלמות.

טו. שני מוצרים הם פועל יוצא מתהליך התכנון של מערכת מידע ממוחשבת — תכניות מחשב ונוהלי עבודה למשתמשים. ללא שני מוצרים אלה אין התהליך מסתיים. חשיבותם של הנהלים אינה נופלת מחשיבותן של התכניות. ללא נהלים כתובים בשפת המשתמש ומפורטים דיים — אין למשתמשים האפשרות להשתמש במערכת המידע. כדי לכתוב את הנהלים יש צורך במפרטים מטעמם של המפתחים, וכן קיים הצורך להדריך את המשתמשים להשתמש במערכת באמצעותם של הנהלים. את הנהלים יש לנסות כפי שמנסים את התכנית.

טז. פעולת ההסבה מהמערכת הקיימת למערכת החדשה היא מערכת בפני עצמה, שלגביה יש צורך בתכנון ובבקרה, בכתיבת תכניות, בנהלים, בהדרכה וכד׳, כלומר, קיימות למעשה שלוש תת־מערכות: המערכת הקיימת, מערכת ההסבה והמערכת החדשה. ההחלטה לעבור משלב הפיתוח לשלב ההסבה היא החלטה קריטית. אין לעבור לשלב ההסבה לפני שיתברר לחלוטין שתהליך הפיתוח אמנם הסתיים במלואו.

יז. כל ארגון העוסק בפיתוחן של מערכות מידע ממוחשבות חייב לפתח לעצמו תכנית־אב. מטרותיה של תכנית האב הן:

— להקיף את תכנית הפיתוח לטווח הבינוני והארוך ובאמצעותה להכין את המשאבים הדרושים לביצועה
— לקבוע את סדר הפיתוח האופטימלי
— לפתח את המערכות כך, שבסופו של דבר ניתן יהיה לשלב מערכות במקום שהדבר נדרש
— למנוע פיתוח ״פראי״ של מערכות מידע ללא קשר לתכנית כוללת, הלוקחת בחשבון את צורכי הארגון ומטרותיו.

המשתנים העיקריים שיש לקחת בחשבון בעת קביעת סדר העדיפות של פיתוח המערכות השונות בתכנית האב הם:
— הסדר הלוגי של שלבי הפיתוח
— יכולת הארגון לקלוט את המערכת
— המשאבים הדרושים לפיתוח
— האם המערכת עונה לצרכים בשטחים החשובים לארגון
— התמורה הגלומה (הישירה והעקיפה) לארגון מפיתוחה של המערכת
— אילו מערכות פועלות ומפותחות אצל ארגונים מתחרים וארגונים קשורים

— יכולת הפיתוח של הארגון (כוח־אדם מקצועי, טיבו וכמותו).
— היבטים פסיכולוגיים (הצורך ברכישת אמונו של הארגון על־ידי היחידה המפתחת את המערכת)

ניתן להבחין בקבוצת המשתנים המפורטת לעיל בין מדדים כמותיים לבין מדדים איכותיים. יש להגדיר מראש שיטה לשימוש מושכל במשתנים.

יח. בתהליך הפיתוח של מערכות מידע ממוחשבות צריך לקחת בחשבון את צרכיו ויכולתו של המשתמש, ובה בשעה גם את צרכיה ויכולתה של היחידה שתתפעל את המערכת. היחידה המתפעלת היא בחינת בית־חרושת שפעולותיו צריכות להיות מוגדרות, שגרתיות וחד־משמעיות ככל האפשר. נוסף על כך, מערכת מידע ממוחשבת עוברת לאחר גמר פיתוחה שינויים ותוספות, והיא צריכה להיות ערוכה לביצועם בצורה המיטבית.

יט. לא כל רכיביה של מערכת מידע הם ממוחשבים. כל מערכת כזו מורכבת מרכיבים ממוחשבים ומרכיבים ידניים, הקשורים ביניהם ומהווים יחדיו מערכת אחת הרמונית. משמעות העובדה, שפיתוח מערכות המידע חייב בו־זמנית לכלול פיתוח הן של רכיבים ממוחשבים והן של רכיבים ידניים.

כ. חוק התפוקה השולית הפוחתת תקף בכל הקשור לפיתוחן של מערכות מידע. תוספת עובדים לצוות הפיתוח עשויה לתרום משמעותית לקצב הפיתוח, אבל רק עד לגבול מסוים. הגדלתו של הצוות אל מעבר לגבול זה אינה מיטבית והתפוקה השולית שלו תפחת ככל שיגדל הצוות. במלים אחרות, הגדלתו של הצוות העוסק בפיתוח אינה תורמת בהכרח להחשת קצב הפיתוח.

תהליך הפיתוח והיישום — פעילויות ומבצעים

תהליך הפיתוח והיישום של מערכות מידע ממוחשבות מתחלק כאמור לארבעה שלבים עיקריים, ואלה מצדם מתחלקים לשלבי־משנה כפי שיפורט להלן. פיתוח המערכות חייב להיעשות שלב אחר שלב בצורה שיטתית וללא פסיחה על שלבים. המשקל הסגולי של השלבים השונים אפשר שישתנה מארגון לארגון וממערכת למערכת, אולם אין לשנות את הסדר ואת השלמות הצריכים לאפיין את פיתוחן של כל המערכות. התהליך הלוגי המתואר להלן מתאים הן לפרוייקטים גדולים הכוללים רכישת מערכת חומרה גדולה ופיתוח עצמי של תוכנה, והן לפרוייקט מיחשוב מערכת קטנה מאוד שתתבסס על ציוד קיים או על רכישת מחשב זעיר ורכישת חבילת תוכנה מוכנה. ברור שהפעילויות שיבוצעו בכל שלב ותשומות כוח־האדם בכל שלב יהיו שונים מפרוייקט אחד למשנהו.
לכל שלב הפונקציה המבצעת אותו (מנתח מידע, מתכן מערכת, תכניתן וכו׳). ייתכנו מקרים שבהם מספר פונקציות יבוצעו בידי אותו אדם. מנתח המידע ומתכן המערכת יכולים למשל להיות אותו אדם, כאשר הפרוייקט קטן בהיקפו.

אחת מבעיות היסוד של תהליך הפיתוח הנה הממשק (interface) בין השלבים, במיוחד כשהאחריות לביצוע עוברת לאדם אחר (לדוגמה, במעבר משלב תיכון המערכות לשלב התכנות). האחראי על הפיתוח חייב לוודא מקרוב, שאכן המעבר נעשה באופן חלק ושלם ושכל המידע הועבר מהאחראי האחד לשני, הן בכתב (תיעוד) והן בעל־פה. קיימה אפוא תקופת מעבר בין שלב אחד למשנהו, שבה עובדים האחראים על השלבים בצוותא כדי לשמור על רציפותו של תהליך הפיתוח.

כל שלב יוצר פלט המשמש כקלט לשלב הבא אחריו. לפני הפיכתו לקלט חייב הפלט לעבור בדיקה שתקיף את הנקודות האלה:

— ההיבט התיפקודי של המערכת

— היישימות הכלכלית, הטכנית והארגונית

— הצד התכנוני — לוח זמנים עתידי, המשאבים הדרושים ויכולת הקצאת המשאבים להמשך הפיתוח

לסיכום, פיצול תהליך הפיתוח למספר שלבים ולתת־שלבים מאפשרת בקרה על התהליך הנעשית בעזרת קטעים בני־שליטה (מבחינת ההשקעה, המוצר ולוח זמנים), האצלת אחריות וסמכות לבעלי המקצוע המתאימים, ביצוע איטרציות דרושות בצורה מיטבית ובדיקת היישימות של המערכת באופן פרוגרסיבי ובר־שליטה.

מחזור החיים של מערכת ממוחשבת בנוי מהשלבים האלה:

הגדרה

1. ניתוח ראשוני — הגדרת הדרישות למידע וקביעת היקף מערכת המידע וגבולותיה

2. חקר ישימות — קביעת ישימותה של המערכת המוצעת מהבחינות: הכלכלית, הטכנית והארגונית

3. ניתוח המידע — בניית המפרטים התפקודיים המפורטים של המערכת החדשה וכן קביעת דרך המעבר (ההסבה) מהמערכת הקיימת לחדשה.

4. תיכון המערכת — תיכון המפרטים לתכנות, להסבה ולכתיבת נהלים.

בנייה

5. תכנות — קידוד התכניות וניסויין
6. כתיבת נהלים

התקנה

7. הסבה — ניסוי המערכת החדשה והבאתה לפעולה תקינה

הפעול

8. תפעול ואחזקה
9. בקרה — בקרה שוטפת ושיטתית על המערכת
10. סיום — החלטה לחסל את המערכת או להחליפה בחדשה.

חלוקה זו לשלבים היא אחת מני רבות המוצעות בספרות. עם זאת, כל החלוקות שוות בעקרון וההבדלים בעיקרם הם הבדלים ברמת הפירוט.

להלן נתונים אופייניים על פילוג ההשקעה ועל זמן הפיתוח באחוזים בשלבי הפיתוח השונים של מערכת מידע ממוחשבת (חלק מהשלבים ניתן לביצוע בחפיפה ולכן פילוג זמן הפיתוח המוצג כאן עשוי להיות מוטה):

השלב	ההשקעה	זמן הפיתוח
ניתוח ראשוני	2%	5%
חקר ישימות	1%	5%
ניתוח המידע	10%	15%
תיכון המערכת	8%	10%
תכנות וכתיבת נהלים	40%	35%
הסבה — התקנה	39%	30%
	100%	100%

טבלה 5.1 מתארת את השלבים ומטרותיהם, מי מבצע כל שלב, מי מאשר את המעבר משלב אחד למשנהו ומהו התוצר של כל שלב.

טבלה 5.1 : שלבי מחזור החיים של מערכת מידע ממוחשבת

פעילות	תוכן	מבצעים	אישור מעבר לשלב הבא	התוצר
1. ניתוח ראשוני	בחינת צורכי המידע, קביעת גבולות המערכת, הגדרת מקום המערכת בארגון, אפיון זרימת הנתונים, הערכת נפחי הנתונים, ניתוח מערכת נוכחית מול צרכי המידע	מנתח מידע משתמש	משתמש והנהלה	מפרט דרישות בשפת המשתמש
2. חקר ישימות	בדיקת ישימות כלכלית, טכנולוגית, ארגונית	מנתח מידע משתמש	משתמש והנהלה	דוח חקר ישימות בשפת המשתמש
3. ניתוח המידע (תיכון לוגי)	1. תכנון הפרוייקט 2. תכנון מפרטים לוגיים של המערכת. 3. משאבים דרושים	מנתח מידע משתמש	הנהלה	דוח ניתוח מערכות בשפת המשתמש
4. תיכון המערכת (תיכון טכני, פיסי, מפורט)	1. בדיקת ישימות (חוזרת) 2. בחירת תוכנה, חומרה.	מתכנן (מעצב) מערכות	מנהל פיתוח מערכות, משתמש, הנהלה	תיקי בחינה — מפרטי התכניות מפרטי נהלים מפרטי ניסוי
5. תכנות	קידוד, ניסוי, תיעוד	מתכנתים	מתכנת ראשי	תכניות, תיעוד

פעילות	תוכן	מבצעים	אישור מעבר לשלב הבא	התוצר
6. פיתוח נהלים	כתיבת וניסוח נהלים למשתמשים, לבקרה, למפעילים ולצוות ההסבה.	משתמש, מתכן מערכות, מנתח	משתמש	נהלים (מדריכים)
7. הסבה	הסבת תכניות, קבצים ונהלים, מבחני קבלה (ניסוי המערכת בכללותה) ע״י המשתמש	הנ״ל + מפעילים ומתכנתים	משתמש והנהלה	יומן ההסבה; מערכת מוכנה לתפעול
8. תפעול ואחזקה	תפעול המערכת, טיפול בשינויים, אחזקה	משתמשים מנתחי מידע מתכן מערכות מתכנתים מפעילים		מערכת מידע פועלת, יומן שינויים יומן אחזקה
9. ביקורת	סקירת מטרות ומימושן, עלות־תועלת של המערכת	משתמש, מנתח מידע, מבקר ענ״א	משתמש והנהלה	דוח ביקורת
10. סיום	בדיקה לקראת החלטה על מחזור חיים חדש	משתמש, מנתח מידע	הנהלה	ממצאי הבדיקה וההחלטה

פירוט שלבי הפיתוח (ראה תרשים 5.2)

1. התוצאה של שלב ניתוח הדרישות הראשוני תכיל את הנתונים דלהלן:
   — פירוט צורכי המידע
   — הגדרת המערכת
   — זרימת המידע (מקורות ויעדים)
   — תדירות קבלת המידע
   — תיאור העיבודים הנדרשים
   — היקפה וגבולותיה של המערכת שתפיק את המידע.

2. שלב חקר הישימות חייב לספק תשובות לשאלות האלה:
   — מהן המטרות אותן אמורה המערכת להשיג
   — כיצד יושגו המטרות — פתרונות רעיוניים (קונספטואליים) אפשריים

   בחינת כל חלופה (פתרון רעיוני) במישורים האלה:
   — האם ניתן לפתח את המערכת מהבחינה הטכנית
   — איך תתקבל המערכת על־ידי המשתמשים המיועדים — ישימות ארגונית
   — הכדאיות הכלכלית של המערכת — האם היחס בין עלות לתועלת מצדיק את פיתוחה של המערכת — ישימות כלכלית
   — האם תואמת המערכת את תכנית האב ואת סדרי העדיפויות שקבעה ועדת ההיגוי (או הנהלת הארגון).

   מסקנת הבדיקה צריכה להיות המלצה על החלופה האופטימלית לאור בדיקת הישימות, ופירוט השלבים הבאים הנדרשים.

3. בשלב ניתוח המידע ייבנו המפרטים התפקודיים המפורטים של מערכת המידע. המערכת שהוגדרה בשלב הניתוח הראשוני תפורק למרכיביה (לחוליות או לתת־מערכות) והקשרים ביניהם יוגדרו במפורט. כמו־כן יתבצע תיכון של המערכת ברמת פירוט המאפשרת את בדיקת יכולתה של המערכת להיות מיושמת טכנית.

   ולבסוף, תיעשה בדיקה חוזרת בהיבטי הישימות השונים של המערכת. משום היקפו וחשיבותו מתחלק שלב זה לשלושה: הבנת המערכת הקיימת, קביעת צורכי המידע ותיכון לוגי של המערכת החדשה שתספק את הצרכים הללו.

4. בשלב תיכון מערכת המידע הממוחשבת יתבצע תרגום המפרטים התיפקודיים שנבנו בשלב הקודם בשפת הלקוח, עבורו ועם המבט אל הארגון, לשפת התכניתן ועם המבט אל המחשב. המרכיבים שאותם הוחלט למחשב יקבלו את עיצובם הממוחשב האופטימלי וייקבעו המפרטים לתכנות ולכתיבת הנהלים. בשלב זה תפורק המערכת לתכניות. מפרטים אלו יכילו את הנושאים האלה לגבי כל תכנית:
   — קלט: מקור, תוכן ומבנה הקלטים, כמויות מוערכות

— קבצים: תוכן, מבנה וצורת גישה לקבצי המערכת, נפחים מוערכים
— עיבוד: כללי העיבוד, קצב העיבוד
— פלט: תוכן וצורת הפלטים, כמויות מוערכות
— ניסוי, הסבה והתקנה: נתוני הניסוי, צורת הניסוי ומדדים לבדיקת טיב הניסוי של התכניות, צורת בניית הקבצים החדשים וצורת בדיקת תכנם, שיטת ההסבה ותהליך ההסבה.

תיק הבחינה שהוא התוצר של שלב זה יהיה בנוי בצורה הזו:
— תרשימי זרימה של המערכת ושל התכניות שיתארו את תזרים הקלטים, העיבוד, הפלטים, הקבצים והציוד שבו ישתמשו. יושם דגש על תיאור הקשרים בין התכניות
— טבלאות שיתארו את הקשר בין נתוני הקלט, הקבצים והפלט
— מערך טפסי המקור, קבצים ופלטים
— תיאור כל בדיקות התקינות, הסבירות והשלמות
— תיאור מילולי של כל תכנית, תיפקודה וכללי העיבוד המתבצע בה
— פירוט המפרטים לכתיבת נוהלי ההפעלה והשימוש
— נתוני בדיקה של כל תכנית ושל המערכת כולה
— תכנית לבניית הקבצים, באמצעות איסוף נתונים אצל המשתמש או באמצעות הסבתם מקבצים קיימים
— תכנון התקנתה של המערכת
— תהליך הבחירה של חומרה ותוכנה (אם המערכת החדשה מחייבת השגת חומרה ותוכנה מספקים חיצוניים)
— תכנון עבודת התכנות וכתיבת הנהלים (לוח זמנים והקצאת משאבים).

5. בשלב התכנות נכתבות תכניות המערכת בשפת מחשב ועוברות ניסוי. כל מתכנת אחראי לבדיקת התכנית שכתב ולניסויה. לאחר שכל תכניות המערכת נוסו בנפרד, יש לבצע ניסוי של כל מערכת התכניות. ניסוי משולב זה מתמקד בממשקים שבין התכניות הבודדות ומבוצע על־ידי מתכן המערכת או ראש צוות הפיתוח. ניסוי מוצלח פירושו שהמערכת מוכנה לשלב ההסבה ולמבחני קבלה ע״י המשתמש. כאשר מערכת המידע משתמשת בתוכנה מוכנה הנרכשת מספק חיצוני, יתמקד שלב ההכנות בהתאמת התוכנה למערכת, בבדיקתה ובניסויה.

תיעוד התכניות הוא חלק חיוני מפעילות התכנות. תיעוד טיפוסי של תכנית משלים את תיק הבחינה וכולל את התוספות דלהלן:

תיאור מילולי: תיאור מטרת התכנית והפתרון, שהיא מספקת לבעיה מסוימת.

הצגה לוגית: תיאור הצעדים הלוגיים של התכנית (לעתים קרובות בעזרת תרשים זרימה של תכנית).

הדפס התכנית (program listing): עותק מודפס של תכנית המקור המיוצר ע״י המחשב בזמן ההידור.

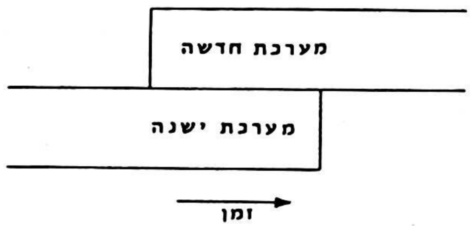
תבניות קלט/פלט: תיאור הנתונים, הרשומות והדוחות המראים את המיקום היחסי של כל שדה בכל רשומה.
נתוני הניסוי: עותק של נתוני הניסוי.
הוראות הפעלה: כל ההוראות הדרושות להרצת התכנית על־ידי המחשב.

6. בעוד שבשלב התכנות נוצרות הוראות למחשב, הרי שבשלב פיתוח הנהלים נוצרות הוראות לרכיבים האנושיים של המערכת החדשה. הנהלים עבור המשתמשים והמפעילים השונים של המערכת נכתבים בשלב זה ומנוסים. כל נוהל חייב לעבור ניסוי ע״י המשתמש באותו נוהל (למשל, מפעיל מחשב, מבקר פנימי וכד׳).
לאחר שהתכניות והנהלים נכתבו ונוסו, מוכנה המערכת להתקנה. ההתקנה פירושה הסבה חלקה מהמערכת הקיימת לחדשה.

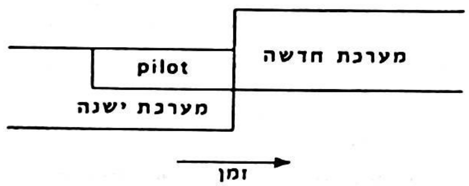
7. בשלב ההסבה יודרכו ויוכשרו מתפעלי המערכת והמשתמשים בה, והמערכת תורץ ותותקן בארגון (שתילת המערכת בארגון).

קיימות 4 שיטות עיקריות לביצוע המעבר מהמערכת הישנה למערכת החדשה:

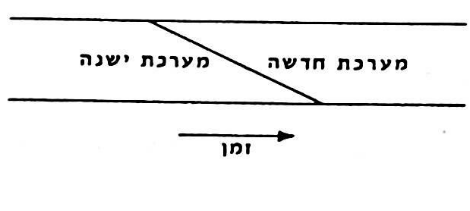
(א) ריצה במקביל



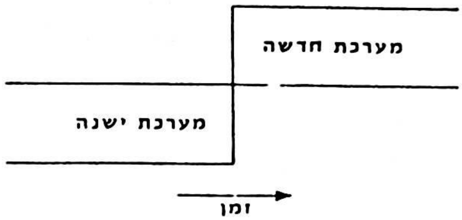
(ב) מעבר דרך מערכת נסיונית (pilot)



(ג) מעבר הדרגתי (phase-in) — חוליה אחר חוליה



(ד) מעבר חד מהמערכת הישנה לחדשה

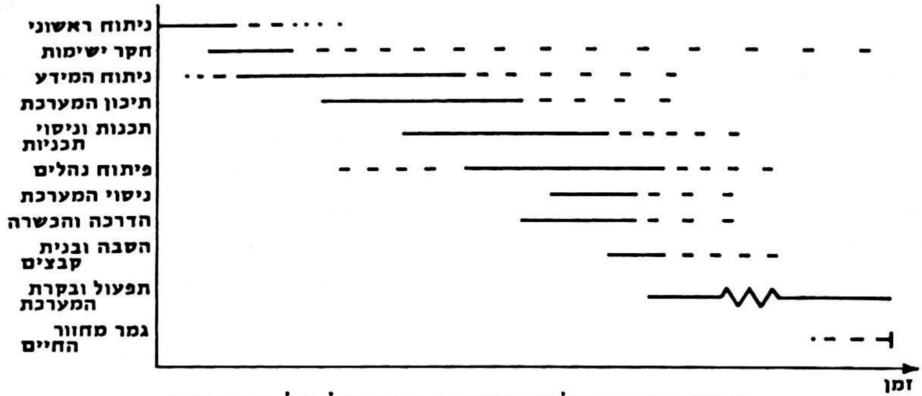


בכל מקרה יש לשקול איזו שיטה היא הטובה ביותר; ייתכן שילוב בין השיטות.

8. לאחר גמר ההסבה פועלת המערכת כמתקן ייצור, מעבדת נתונים ומייצרת מידע. שגיאות שונות שמתגלות ודרישות לשינויים הם למעשה פרוייקטים של פיתוח מערכת בקנה־מידה קטן, עם מחזור חיים דומה:
— ייזום הפרוייקט
— סקר של בעיית אחזקה
— בנייה (ניתוח, תיכון, פיתוח, ניסוי)
— התקנה
— הערכה סופית.
טיפוסי לשלב התפעול הוא מעורבות המערכת עם סוגים חדשים של אנשים בארגון: פעילויות האחזקה מבוצעות ע"י מתכנתי אחזקה; משתמשים; מנתחי מידע ומתכני מערכות מעורבים לעתים קרובות בפרוייקטים של אחזקה; מערכת המידע מופעלת על־ידי מפעילי מחשב ועל־ידי אנשי קלט/פלט.

9. במהלך תפעול המערכת קיימות נקודות זמן שבהן יש להעריך את האופיינים התפעוליים של המערכת, את עלויות התפעול ואת השיפורים הנחוצים כדי שהמערכת תפעל ביתר יעילות. בקרה זו, המתבצעת באופן שוטף ותקופתי, מגלה בעיות או פערים, שאפשר שהם ניתנים לפתרון באמצעות אחזקה שוטפת, או אפשר שהם רציניים יותר ומצביעים על התיישנות המערכת. המטרה הבסיסית של בקרה זו היא להשוות את המטרות, לאמוד את היקף העלויות ואת התועלות החזויות של המערכת, כפי שנקבעו בתהליך הפיתוח, עם הניסיון התפעולי.

10. סיום חייה של המערכת (ובדרך־כלל גם תחילת חייה של מערכת חדשה — שמכאן המושג "מחזור החיים") בא לידי ביטוי באותו שלב שבו הדרישות לשינויים מחד ומספר השגיאות במערכת מאידך, מגיע לרמה כזו שלא כדאי להמשיך ולהפעיל עוד את המערכת הקיימת.



תרשים 5.2: שלבי הפיתוח והתפעול על פני הזמן

פרוייקט הפיתוח של מערכת מידע ממוחשבת

שלבי פרוייקט הפיתוח של מערכת מידע ממוחשבת אינם מתבצעים דווקא בטור. מספר שלבים יכולים להתבצע במקביל. מאידך, ישנם שלבים שבהם לא ניתן להתחיל קודם שהסתיים השלב שלפניהם. ביצוע השלבים על פני זמן מתואר בתרשים 5.2.

יש להזכיר שוב, שהתיעוד הוא מוצר של תהליך הפיתוח. הוא נוצר בעת תהליך הפיתוח וסיום כתיבתו מסמל את סיום שלב הפיתוח (כל שלב והתיעוד שלו). מבחינים בארבעה סוגי תיעוד עיקריים:

א. מפרט המערכת המתאר את המערכת כולה — נכתב בידי מומחי מערכות מנתח המידע (analyst) ומתכן המערכת (designer).
מפרט המערכת מכיל מספר סוגים משניים של תיעוד:
קונספציה — תיאור מערכת המידע בשפתו של המשתמש ועבורו.
תיק־בחינה — תיאור מערכת המידע על רכיביה (התכניות, מאגרי הנתונים ומהלך הזרימה) בשפתו של התכניתן ועבורו.

ב. תיעוד התכניות המכיל את פרטי כל תכנית ותכנית במערכת — נכתב בידי המתכנתים.

ג. תיעוד למפעילי המערכת — נכתב בידי התכניתנים, מנתחי המידע והמשתמשים.

ד. הנהלים למשתמשים במערכת — נכתבים בידי מנתחי המידע ובשיתופם הפעיל של המשתמשים.

מערכת מידע ממוחשבת הנה כלי בידי הארגון שעבורו היא נבנית. בארגון היא משמשת כלי הן בידי המערכת התפעולית והן בידי המערכת הניהולית. מערכת המידע חייבת להשתלב בארגון, ליהפך לחלק ממנו ולענות על צרכיו של הארגון. כמו־כן מדובר ב"מוצר" שלשם יצירתו הושקעו משאבים רבים.
המסקנה בעקבות האמור לעיל היא, שקיים צורך במעורבותה הפעילה של הנהלת הארגון בתהליכי הפיתוח והיישום של מערכות המידע. בעוד שהנהלת הארגון צריכה להיות מעורבת בעיקר בתמונה הכללית של התהליכים הללו, נדרשת מעורבות פעילה יותר מצד המשתמשים שעבורם נבנית המערכת. כדי שמעורבות ההנהלה והמשתמשים תבוא לידי ביטוי בצורה יעילה ושיטתית, יש צורך בקביעת מסגרות ארגוניות מתאימות, שיאפשרו את המעורבות הדרושה. מסגרות אלו כוללות את הגופים הארגוניים האלה: ועדת מערכות מידע (שהיא גוף קבוע), קבוצת חקר ישימות, ועדת היגוי, קבוצת הפיתוח וקבוצת ההתקנה.

ועדת מערכות מידע

ועדה זו משמשת כמועצת מנהלים לראש יחידת מערכות המידע בארגון. חברי הוועדה שייכים לרמת הניהול הבכיר בארגון. חברים טיפוסיים יכולים להיות, למשל, הסמנכ"לים למימון, לשיווק, לייצור, לכוח־אדם וכן ראש יחידת מערכות מידע והמבקר הפנימי. הוועדה יכולה להיפגש פעם בחודש. היא יוזמת ביצוע חקר ישימות מקדים למערכות מידע חדשות שהוצעו, סוקרת דוחות ישימות, מאשרת או דוחה פרוייקטים של פיתוח, סוקרת דוחות בקרה תקופתיים, קובעת קדימויות פיתוח ומאשרת את תכנית האב או שינויים בה. לוועדה זו האחריות הכוללנית לפיתוח מערכות המידע בארגון ולתפעולו, במסגרת הסמכות והמשאבים המוענקים מידי הנהלת הארגון.

קבוצת חקר ישימות

קבוצה זו פועלת זמנית לצורך בדיקת הישימות של פיתוח מערכת מידע חדשה מסוימת. בדרך־כלל כוללת הקבוצה לפחות שלושה חברים: נציג המשתמש העיקרי בארגון, שאותו תשרת המערכת כאשר תותקן, נציג ההנהלה הכללית של הארגון ונציג יחידת מערכות המידע (בדרך־כלל יהיה זה מנתח מידע בכיר). הרכב זה משקף את הגופים הנוגעים לפיתוח המערכת ומבטיח מנגנון של בקרה ואיזון בתהליך חקר הישימות. במקרים רבים תורחב הקבוצה, בייחוד כאשר קיימים מספר משתמשים עיקריים, או כאשר יש צורך להיוועץ במומחה בתחום מיוחד. בדרך־כלל יהיה ראש הקבוצה נציג המשתמש העיקרי.
דוח חקר הישימות מוגש לוועדת מערכות מידע שיכולה להחליט על אחת מהאפשרויות האלה:

— להתחיל בפרוייקט הפיתוח
— לנטוש את הצעת הפיתוח (זמנית או סופית)

— להמשיך בחקר הישימות (לאסוף מידע נוסף).
אם ההחלטה היא להתחיל בפרוייקט הפיתוח, יש להקים את ועדת ההיגוי ואת קבוצת הפיתוח.

קבוצת הפיתוח

מטרת הקבוצה היא לנהל את פרוייקט פיתוח מערכת המידע עד לנקודה שבה המערכת מוכנה להתקנה בארגון (מנקודה זו אחראית לנושא קבוצת ההתקנה). בראש הקבוצה עומד מנתח מידע בכיר (שיתכן שהיה גם חבר בקבוצת חקר הישימות). חברי הקבוצה הם אנשי מקצוע המועסקים ביחידת מערכות המידע של הארגון או של ארגון חיצוני. בארגונים שבהם יחידות פונקציונליות (מימון, שיווק וכו') מעסיקות מומחי מידע, רצוי שאחד מהם לפחות יצורף לקבוצה.
הרכב הקבוצה משתנה עם הזמן, בהתאם להתקדמות בשלבי מחזור החיים של המערכת המפותחת. תרשים 5.3 משקף הרכב אופייני של קבוצת פיתוח בארגונים גדולים או בפרוייקט גדול.



תרשים 5.3: מעורבות מומחי מידע שונים בקבוצת הפיתוח של מערכת

ועדת היגוי

תפקידה העיקרי של ועדה זמנית זו הוא לוודא שמערכת המידע תואמת את דרישות המידע שהניעו את הארגון ליזום את תהליך הפיתוח. הוועדה מלווה את הפרוייקט לכל אורכו. תפקידה יכול להסתיים לאחר שהמערכת הותקנה, אך גם לאחר מכן היא מתכנסת לתקופה קצרה כדי להעריך את דוח הבקרה הראשוני על המערכת המופעלת. אחריות ועדת ההיגוי מקיפה את תכנון הפרוייקט, ארגונו ובקרתו.
הוועדה מורכבת ממנהלים בכירים של הארגון ויכולה להיות תת־ועדה של ועדת מערכות המידע של הארגון. ההרכב חייב לכלול מנהל בכיר המייצג את ההנהלה הכללית, מנהל(ים) בכיר(ים) המייצג(ים) את המשתמש(ים) העיקריים במערכת וראש יחידת מערכות מידע. הצורך בייצוגה של הנהלת הארגון בוועדת ההיגוי, נובע מכך שגבולותיהם של מערכות מידע אינם תואמות בהכרח את המבנה הארגוני הפורמלי. ניתן לצרף לוועדה יועץ חיצוני בעל ניסיון וידע שיוכל לשמש בתפקיד מבקר פנימי. ראש קבוצת הפיתוח משתתף בישיבות הוועדה. לשם יעילות העבודה רצוי שוועדת ההיגוי לא תהיה רבת חברים.
תדירות הפגישות של הוועדה יכולה להשתנות, אולם היא חייבת להתכנס ולסקור

את שלבי ההתקדמות העיקריים, בייחוד לפני שהפרוייקט נע משלב אחד במחזור החיים לשלב הבא אחריו.

תפקידיה העיקריים של ועדת ההיגוי:

— אישור ושינוי יעדי ביניים ויעדים סופיים בתהליך הפיתוח
— אישור ביצועים מוצעים של המערכת המפותחת (למשל זמני תגובה, תדירות דיווחים)
— הגדרת הקריטריונים שלפיהם תוערך הצלחת (או כישלון) המערכת לאחר התקנתה
— אישור רכישות עיקריות של תוכנה וחומרה למערכת
— אישור מינויים של אנשי מפתח בפרוייקט (יועצים חיצוניים, ראשי קבוצות פיתוח והתקנה)
— "מכירת" המערכת להנהלה הכללית ולהנהלה של מחלקות משתמשות בארגון
— סקירה תקופתית של דוחות התקדמות
— אישור תכנית הפרוייקט (לוחות זמנים, תקציב)
— קביעת קדימויות, בייחוד כאשר המקורות הזמינים לביצוע הפרוייקט מכתיבים חלוקת הפיתוח לחוליות
— סקירת דוחות ישימות
— אישור תוצאות ניסוי המערכת
— סקירה של כל דוח עיקרי שנוצר בתהליך הפיתוח (מפרט המערכת, תיק בחינה, תכנית ההסבה) ואישור לעבור לשלב מתקדם יותר בתהליך
— תיאום ההסבה וההתקנה
— סקירת דוח הערכת המערכת לאחר ההתקנה.

קבוצת ההתקנה

קבוצה זו מהווה המשך טבעי לקבוצת הפיתוח וחייבת לכלול חברים נבחרים מתוכה, כגון ראש קבוצת הפיתוח ומספר מתכנים ותכנתים בכירים. כיוון שתפקידה העיקרי לדאוג להתקנתה החלקה של המערכת החדשה, עליה לכלול גם את האנשים האלה:

— משתמשים המייצגים את רמות הניהול ואת רמת התפעול הרלבנטיות למערכת החדשה; על אלה לתכנן ולבצע את פעילויות ההדרכה, ההסבה והניסוי
— מנהלים בכירים (מנהלי מחלקות או משנים להם). השתתפותם בקבוצה חיונית כדי לתמוך בהחלטות העלולות לגרום לשינויים ארגוניים או לשינויים בשיטות עבודה קיימות. (טיפול בהתנגדות האופיינית להכנסת אוטומציה וחידושים בארגון).

בארגונים רבים מטפלת קבוצה אחת גם בפיתוח וגם בהתקנה. אין דופי בסידור מסוג זה, כל עוד חברים חדשים מצטרפים לקבוצה כאשר שלב ההתקנה מתקרב.

גישות לפיתוח מערכות מידע ממוחשבות

קיימות מספר גישות עקרוניות לפיתוח מערכות מידע ממוחשבות. יש להבחין בין הגישות לבניית תכנית האב של מערכות המידע עבור הארגון (או תת־ארגון) לבין הגישות לפיתוח מערכת מידע ספציפית. אבחנה זו אינה עקרונית, כפי שיובהר להלן.

תכנית אב לבניית מערכות היא תכנית כוללנית, המקיפה את צורכי המידע השונים של הארגון, מתחמת אותם למערכות מידע ספציפיות וקובעת את יחסי הגומלין ביניהם.

תכנית האב משמשת לתכנון עבודת פיתוח ליישום מערכות המידע בארגון לטווח הבינוני והארוך. מאחר שאנו מצויים בסביבה של משאבים מוגבלים, מתבצעת עבודת הפיתוח בשלבים, שהוכתבו כפועל יוצא מצרכיו של הארגון, יכולתו לפתח ויכולתו לקלוט ולתפעל את המערכות.

פיתוחה של מערכת מידע ספציפית המהווה חלק מתכנית האב, מתבצע במסגרת כלכלית נתונה, לאמור: גבולותיה של המערכת ויחסי הגומלין שלה עם שאר המערכות נתונים למעשה על־ידי תכנית האב. עם זאת, המבנה הפנימי של המערכת אינו נתון.

גישות לפיתוח תכנית־אב למערכות מידע בארגון

יש להפריד בין גישות שאינן מתאימות (למרות שארגונים לא מעטים עדיין פועלים לפיהן), לבין גישות מתאימות שאימוצן מהווה בסיס הכרחי (אם כי לא מספיק) להצלחת המערכת החדשה.

גישות בלתי מתאימות

גישת אד־הוק

פיתוח מערכות המידע מבוצע ללא תכנון מוקדם, ובהתאם לצרכים המתעוררים חדשות לבקרים. גישה זו של כיבוי שרפות, אינה עושה שימוש בתכנית־אב. תוצאותיה של גישה זו הן "איי־מיכון" ללא התייחסות לקשרי הגומלין ולשילוב אפשרי בין המערכות השונות. יש הצדקה לגישה זו רק במקרים שבהם נמצא הארגון בתהליך שינוי מתמיד.

עם התייצבותו של הארגון, עוברות בדרך־כלל המערכות שנבנו לאור גישה זו תיכון מחדש.

גישת איסוף הנתונים

לפי גישה זו נאספים הנתונים הקיימים בארגון, נרשמים, נאגרים ומעודכנים — מבלי שיבוצע ניתוח מוקדם בכל הקשור לשימוש שניתן לעשות בהם. בהתעורר הצורך במידע מסוים, מתבצע עיבוד הנתונים הדרוש לשם הפקת מידע זה.

מערכות הבנויות בהתאם לגישה מכוונת זו קוראים בשם: input oriented systems. הצידוק לשימוש בגישה זו הוא הטענה שלא ניתן לחזות את צורכי המידע של הארגון, והדרך היחידה האפשרית היא אפוא איסוף הנתונים ואגירתם, וכן אספקת כלים לעיבודם ולשליפתם בעת הצורך.
גישה זו אינה מעשית משום שלא ניתן לאסוף, לאגור ולעדכן את כל נתוני הארגון, ומשום שהמשתמש חייב בכל מקרה שבו הוא זקוק למידע לדרוש עיבוד מיוחד כדי להפיק אותו.
גם גישה זו אינה עושה שימוש בתכנית־אב.

גישת התרשים הארגוני

גישה זו מניחה שמערכות מידע חופפות בדרך־כלל את מבנה הארגון על יחידותיו השונות. גישה רווחת זו יוצרת קשיים מעשיים רבים. פיתוח עצמאי של יישומים נפרדים מביא להכפלת מאמצי התיכון. דרישות המידע והעיבוד של הארגון אינן זהות תמיד לגבולות הארגוניים הקיימים. למשל, ביצוע מכירה גורם לתנועות במערכת המכירות, בחשבון חייבים, במלאי מוצרים מוגמרים וביומן כללי, ובעקיפין גם לתנועות במערכות הייצור והרכש. למערכת מידע התפורה בדיוק לפי גבולות ארגוניים קשה לטפל בפעילויות בין־מחלקתיות ולספק מידע שנדרש על־ידי מחלקה אחרת בארגון.
לפי גישה זו, תכנית האב היא ברמת היחידה הארגונית ולא ברמת הארגון כולו.

גישות מתאימות

הגישה מלמטה למעלה (bottom up)

לפי גישה זו נבנית תכנית האב בהתאם לאירועים ולפעולות המתבצעים בארגון. כלומר בשלב הבנייה הראשון נבנות מערכות מידע שיפיקו מידע לתפעול בלבד, ורק בשלבים מאוחרים יתפתחו המערכות כך שיוכלו להפיק גם מידע לניהול. שלבי הביצוע לפיתוח מערכות המידע בארגון בגישה זו הם:

1. פיתוח מערכות מידע מבודדות בעלות קבצים יחודיים. המידע המופק הוא לתפעול.
2. שילוב הקבצים היחודיים, במקום שנדרש, למסד נתונים הנתמך ע"י מערכת לניהול מסדי נתונים.
3. הוספת מודלים לתכנון ולעשיית החלטות שיאפשרו הפקת מידע לניהול לדרגי הניהול התיכון (טקטי) והתפעולי.
4. שילוב המודלים לתכנון ולעשיית החלטות למסגרת של מסד מודלים (model base), כך שיוכלו לעמוד לרשותן של כל מערכות המידע

וכן הרחבת מסד הנתונים עם נתונים חיצוניים. בשלב זה יכולות מערכות המידע לתמוך גם בפעילויות הניהול של דרג הניהול הבכיר (אסטרטגי).

גישה זו מאפשרת התאמת המערכות לצרכים הממשיים של הארגון בשיטת הצעד אחר צעד. עם זאת, מגבלתה העיקרית נעוצה בחוסר יכולתה לראות מראש את הצרכים הכוללים של הארגון, לפיכך שילוב המערכות אינו מיטבי. אימוצה של גישה זו מביא בהכרח לצורך בתיכון מחדש של מערכות המידע, משום שהתיכון הראשוני אינו יכול לקחת בחשבון את ההיקף העתידי של המערכות. צרכים שאינם מועלים על־ידי הצרכנים אינם נענים.

הגישה מלמעלה למטה (top down)

לפי גישה זו מאתרים את צורכי המידע הכוללניים של הארגון, ובהתאם להם בונים את מערכות המידע. איתור הצרכים הכוללנים לפני בניית מערכות המידע מאפשר תיחום המערכות ושילובן. שלבי הביצוע לפיתוח מערכות המידע בארגון בגישה זו הם:

1. הגדרת מטרות הארגון.
2. ניתוח סביבת הארגון ואילוציה.
3. הגדרה וניתוח של הפעילויות ושל התהליכים המתבצעים בארגון, לאור מטרותיו.
4. איתור החלטות המפתח לסוגיהן השונים, אשר יש לקבלן במסגרת הפעילויות והתהליכים.
5. איתור המידע הדרוש לתהליכי קבלת ההחלטות ולביצוע הפעילויות.
6. הגדרת יישומים (מערכות מידע ספציפיות), הגדרת קשרי הגומלין שביניהם וקביעת קדימויות לפיתוחם (זו תכנית האב).

הגישה במקביל (parallel)

גישה זו מנסה להתגבר על מגבלותיהן של הגישות "מלמטה — למעלה" ו"מלמעלה למטה", ולשמור במידת האפשר על מעלותיהן.
לפי גישה זו המערכות המספקות מידע ניהולי ומערכות המספקות מידע תפעולי מפותחות ומופעלות בשלב הראשון באופן עצמאי. בעת הפיתוח נעשה מאמץ להשתמש במידע שהוא נגיש ממערכות המידע התפעולי כקלט למערכות המידע הניהולי. באותו זמן מפותחות מערכות מידע ניהולי כדי לנצל את המידע המופק במערכות המידע התפעולי שפותחו.
כאשר גישה זו מיושמת בצורה עקבית, שני סוגי המערכות יתקרבו האחת לשנייה עד שיהפכו לבסוף למערכת מידע כוללת. לגישה זו הולם השם "מהחוץ־לפנים", כיון שהפיתוח מתחיל באופן עצמאי בשתי הקצוות,

והמערכות שפותחו כך עוברות אבולוציה מבוקרת, כל אחת בכיוון רעותה.

הגישה הטובה ביותר

אף לא גישה אחת משלושת הגישות היא הטובה ביותר לכל הארגונים. לכל גישה מעלות וחסרונות. תכנית האב למערכות מידע יכולה להישען על כל גישה מתאימה. למרות שהגישות "מלמעלה-למטה" ו"מלמטה-למעלה" הוצגו כחלופות, הן אינן מוציאות אחת את השנייה; הגישה במקביל משלבת, למעשה, את שתיהן.

כל אחת מהגישות המתאימות שהוצגו לעיל יכולה לשמש בסיס לתכנית אב מוצלחת למערכות מידע, אם היא תואמת את התנאים המיוחדים של כל ארגון. הגישה שיש לאמץ תלויה במבנה הארגון, במורכבותו, בגודלו, בסביבתו, בפילוסופית הניהול, במעורבות ההנהלה הבכירה בתהליכי התכנון והפיתוח של מערכת המידע ובנסיון הארגון עם מערכות מידע ממוחשבות.

גישות לבניית מערכת מידע ספציפית

המושג "גישות לבניית מידע ספציפית" משמש להגדרת צורכי המידע שאותם צריכה מערכת זו לספק. קיימות שתי גישות עיקריות:

— ניתוח החלטות

— ניתוח נתונים.

אלו דומות בעיקרן לגישות "מלמעלה למטה" ומ"למטה למעלה". למעשה, אין גישות אלו עומדות כל אחת בפני עצמה. שילובן הוא שיטת בנייה מעשית ההולמת את מרבית המקרים.

גישת ניתוח החלטות

גישה זו מנסה לאתר את צורכי המידע אשר צריכה מערכת מידע מסוימת לספק על-ידי ניתוח תהליכי קבלת ההחלטות שבהן אמורה המערכת לתמוך. דרך הניתוח היא כדלהלן:

1. זיהוי ואיתור ההחלטות
2. הגדרה וניתוח של תהליכי קבלת ההחלטות (כולל המודל או האלגוריתם שבו עושים שימוש)
3. זיהוי המידע הדרוש לתהליך
4. קביעת הנתונים וצורת עיבודם.

גישת ניתוח נתונים

גישה זו מנסה לקבוע את צורכי המידע שאותם אמורה מערכת מידע מסוימת לספק על-ידי ניתוח הנתונים והמידע שיהיו קיימים והקיימים בארגון. דרך הניתוח היא כדלקמן:

1. איסוף מסמכים, דוחות, קבצים וכד׳, שעושים בהם שימוש בארגון, וזיהוי הנתונים והמידע הכלולים בהם
2. ריאיון צרכני המידע וניתוח מערכות מידע דומות שבהן עושים שימוש בארגונים אחרים, במגמה לאתר נתונים ומידע נוספים פוטנציאליים, שהארגון עשוי לעשות בהם שימוש.
3. תיכון מערכת מידע שתספק את הנתונים והמידע שזוהו והוגדרו.

גישה זו משמעותה התבססות על הצרכים כפי שהם באים לידי ביטוי בהווה על-ידי בעלי התפקידים בארגון ולא כפועל יוצא ממטרותיו ופעילויותיו של הארגון.

תהליך בחירת מערכת ממוחשבת

תהליך הערכת מערכת שונה מארגון לארגון, בהתאם לניסיון של הארגון בשטח המחשבים, הדחיפות בהפעלת המערכת הנבחנת וגורמים אחרים.
גישה נכונה תכלול בדרך-כלל את השלבים האלה:

1. התארגנות
2. לימוד ראשוני
3. בדיקת ישימות
4. הכנת מפרט דרישות
5. גיבוש רשימת ספקים
6. הפצת המפרט
7. בחינת הצעות
8. מו״מ להתקשרות וחתימת חוזה
9. מעקב ביצוע.

1. **התארגנות**

בשלב ההתארגנות מגובשת קבוצת העבודה, נקבעות המטרות והמסגרות של הקבוצה. קבוצת העבודה אחראית לביצוע ההערכה; היא מורכבת בדרך-כלל ממנהלים המייצגים את הפונקציות השונות בארגון וכן מנציג בכיר של הפונקציה הארגונית שאותה תשרת המערכת הנבחנת. רצוי שלחברי הקבוצה יהיה ידע בסיסי במערכות מידע ממוחשבות. נוסף על כך, יש להסתייע בייעוץ מקצועי מהארגון או בייעוץ חיצוני.

משהוגדרו מטרות הקבוצה ומסגרותיה נקבעו למעשה מה רוצים לבחון ומהו מרחב הבדיקה. כלומר — באיזה תחומים ארגוניים (מערכות) מדובר, מהם האילוצים (זמן, תקציב וכד׳), ואיזה נושאים אחרים הם בעלי חשיבות לתהליך הערכת המערכת. חשוב להדגיש כי ללא מטרות ומסגרת אין קבוצת העבודה מסוגלת לתפקד.

2. לימוד ראשוני
כדי לבחון נכונה את הצורך במערכת חדשה, על קבוצת העבודה להכיר היטב את מערכות המידע של הארגון. בשלב זה נבחנות המערכת הקיימת והמערכות הנלוות מבחינת תהליכי יצירת וזרימת הנתונים והמידע, מאגרי הנתונים, שיטות ונוהלי עבודה, צורכי המידע של כל פונקציה ארגונית הקשורה למערכת וכד׳. בתום שלב הלימוד אפשר לקבוע מהו המידע שנדרש, ואיזה חלק ממנו אינו מסופק בצורה נאותה על־ידי המערכת הקיימת.

3. בדיקת ישימות
מטרת בדיקת הישימות להעריך את הצורך בהכנסת מערכת התוכנה והחומרה או את הצורך בהחלפתה, לבחון חלופות שונות מבחינת היעילות, ההשפעה על הארגון והעלות ולהמליץ על החלופה הטובה.
קבוצת העבודה ״מרכיבה״ מספר מערכות חליפיות, ובוחנת לגבי כל אחת את מידת עמידתה בדרישות הארגון, השפעת החלפת המערכת על הארגון ועל העובדים בארגון, יתרונות וחסרונות החלופה, והעלות הצפויה מיישום החלופה. העלות חייבת לכלול את כל מרכיבי העלות (ראה טבלה 5.1).
מתוך החלופות הללו בוחרת קבוצת העבודה את הפתרון הטוב ביותר, שנבחן ביסודיות מבחינת כדאיות, התאמה והשפעה על הארגון. תוצאות חקר הישימות עם המלצות קבוצת העבודה מוגשות לאישור ההנהלה.

טבלה 5.1: דוגמה לניתוח העלות

אמדן העלות של מערכת מחשב חדשה		
עלות הכנת התצורה הפיזית	xx	
בחינה ותכנות של היישומים	xx	
עלות הדרכת כוח־אדם	xx	
התאמת קבצים והפעלה במקביל	xx	
סה״כ הוצאות חד־פעמיות		xx
אמדן עלות הפעלה שנתית		
הוצאות שכירה (או חכירה) ואחזקה	xx	
שכירת תוכנה	xx	
מתכנתים ומתכננים	xx	
מפעילים	xx	
חשמל, שכר־דירה וכד׳	xx	
סה״כ הוצאות הפעלה		xx
חיסכון שנתי (הוצאות נחסכות פחות הוצאות הפעלה שנתיות)		xx

שיעור תשואה (השער שבו הערך הנוכחי
הנקי של החסכונות השנתיים שווה
להוצאה החד־פעמית) xx%
יתרונות לא כספיים אחרים.

4. הכנת המפרט

המפרט הוא מסמך סיכום פנימי המתאר את דרישות הארגון מהמערכת המתוכננת. חלקים ממנו משמשים לצורך קבלת הצעות מספקי חומרה ותוכנה. טבלה 5.2 מרכזת את הפרקים העיקריים הכלולים במפרט.
המפרט צריך לכלול גם מדדים שלפיהם ייבחנו ההצעות וכן מדדים שלפיהם תבחן המערכת שתוקם.

טבלה 5.2: דוגמה למפרט

1. מבוא ותיאור קצר של המערכת
2. יחידות ארגוניות מעורבות
3. תיאור היישומים
   א. מערכות כלליות
   ב. תת־מערכות
4. דרישות מידע
   א. קבצים
   ב. קלט/פלט
   ג. קלט/פלט מיוחד
   ד. מהלכי זרימת המידע
5. אילוצים על תיכון המערכת
6. מהות העבודה
7. מידע נדרש מהספק (ראה טבלה 5.3)
8. קריטריונים לבחירה
   א. כללי
   ב. שיטות הערכה
   ג. קריטריונים טכניים
   ד. קריטריוני עלות
9. אדמיניסטרציה ומידע ליגלי
   א. הצעות הספקים
   ב. דחיית הצעות
   ג. עלויות הקשורות בהכנת ההצעות
   ד. תוספות ל־RFP
   ה. תאריך אחרון להגשת ההצעות

ו. אפשרות להגשת מספר הצעות
ז. אישור תכולת ההצעות
ח. הצגה על־פה
ט. אחריות קבלן ראשי
י. לוחות זמנים
יא. אפשרות למו״מ על החוזה האחיד
יב. סודיות המידע.
10. נספחים.

5. גיבוש רשימת ספקים

רשימת הספקים כוללת את החברות המסוגלות לענות על דרישות המפרט ולהציע מערכות חומרה ו/או תוכנה ברמה סבירה, ואליהם מועבר מפרט הדרישות לקבלת ההצעות.

הכנת רשימת הספקים מחייבת היכרות טובה עם החברות הנמצאות בענף. פנייה לספקים רבים מדי עלולה לגרום לאבדן זמן ולאבדן שליטה; פנייה לספקים מעטים מדי עלולה לגרום לאבדן הצעות מתאימות וטובות; על־כן יש להקפיד על הכנה נאותה של רשימת הספקים.

הספקים שנבחרו מקבלים את מפרט הדרישות של הארגון ומגישים את הצעותיהם, הכוללות בדרך־כלל את הסעיפים המרוכזים בטבלה 5.3. רצוי לדרוש מהספקים להציג את הצעותיהם במבנה אחיד, כדי לאפשר השוואה וניתוח של ההצעות שיתקבלו.

רצוי גם להדגיש בפני הספקים מהם המדדים לבחינת ההצעות.

טבלה 5.3: דוגמה לסעיפים בהצעת ספק של המערכת

1. תצורת הציוד
   א. יחידות הציוד
   ב. נתוני הפעלה וביצוע של הציוד
   ג. הרכבים אלטרנטיביים
   ד. גמישות המערכת — יכולת הרחבה בעתיד (מודולריות)
   ה. דרישות מיוחדות לגבי המקום ועלויות התקנה שונות
2. תוכנה קיימת, וחבילות תוכנה
3. עלות המערכת המוצעת
   א. עלות רכישה ו/או שכירה
   ב. עלות נוספת עבור משמרות נוספות
   ג. חוזה אחזקת הציוד
   ד. שכירת או רכישת תוכנה.

4. סיוע למערכת
   א. ניתוח מערכות הכלול במחיר: עלות עבור שירותים נוספים
   ב. שירותי תכנות הכלולים במחיר: עלות עבור שירותים נוספים
   ג. סיוע והדרכה ועלויות בהתאם
   ד. תנאי שימוש במערכת אלטרנטיבית בעת־הצורך.
5. תנאים
   א. תאריך אפשרי למסירת המערכת
   ב. תנאי תשלום
   ג. אופציית שכירה/קנייה
   ד. זמן ניסוי הכלול במחיר.
6. ביצועי המערכת עבור היישומים במפרט:
   א. שינויים במערכת המוצגת במפרט
   ב. זמני ביצוע
   ג. שינויים בזמני ביצוע כתוצאה מהוספת פריטי ציוד שונים.
7. מידע נוסף

6. הפצת המפרט

בשלב זה יש ליצור קשר עם הספקים שנבחרו, ולאמת את מידת יכולתם ראת רצונם להשתתף בהגשת ההצעות. המפרט מועבר לספקים ולאחר מכן יש לענות על השאלות שמתעוררות. יש להקפיד שכל מידע נוסף יועבר גם לספקים האחרים המשתתפים בהגשת ההצעות, כדי לשמור על שוויון ביניהם בשלב זה. רצוי להזמין את הספקים לסיור היכרות בארגון, להציג את הנושאים הרלבנטיים ולאפשר להם להתרשם מהסביבה שעבורה עליהם להציע מערכות.

יש לקבוע מועד סופי להגשת ההצעות.

7. בחינת הצעות

קיימות שיטות שונות לניתוח ולבחינת ההצעות. שיטה מקובלת היא שיטת הציונים; בשיטה זו נקבעים המדדים העיקריים והמדדים המשניים שמשפיעים על קבלת ההחלטה. למדדים אלה נקבע משקל יחסי המבטא את החשיבות של המדד בהחלטה. שלב זה נעשה לפני קבלת ההצעות מהספקים. עם קבלת ההצעה מכל ספק, נקבע ציון התאמה — מידת ההתאמה של הצעת הספק לכל אחד מהמדדים. שקלול ציון ההתאמה במשקל היחסי של כל מדד וסיכום התוצאות יקבע את מידת ההתאמה של המערכת המוצעת. השוואת מידת ההתאמה של כל הצעה למחירה, תסייע בקביעת המערכת הטובה מבחינת התאמה/עלות.

רשימת המדדים עשויה להיות שונה מארגון לארגון או ממערכת אחת למערכת שנייה. גם מדדים שונים יקבלו משקל יחסי שונה ממערכת

למערכת; נושאים שהם חשובים במערכת אחת יכולים להיות חסרי חשיבות במערכת אחרת. טבלה 5.4 מדגימה את שיטת הציונים.
בחינת ההצעות חייבת להתייחס למכלול ההצעה — לחומרה ולתוכנה, וכן לגורמים נלווים, כגון לתמיכה טכנית של הספק, למוניטין הספק, ליכולת הרחבת המערכת בעתיד, לזמני תגובה וכד׳.

טבלה 5.4: דוגמה לשימוש בשיטת הציונים

קריטריון להערכה	נקודות אפשרויות	נקודות שקיבלו הצעות שונות א.	ב.	ג.
1. ביצועי המערכת (42%)				
א. ביצועי ציוד	60	60	41	42
ב. ביצועי תוכנה	60	39	43	31
ג. אפשרויות הגדלה והרחבה	55	41	48	30
2. תנאי ספק (25%)				
א. מוניטין של הספק	40	25	32	21
ב. תחזוקה	20	14	18	13
ג. סיוע למערכת	20	12	18	12
ד. העדפות עובדים	25	15	20	5
3. עלות (33%)				
א. עלות שכירה	50	35	23	40
ב. תנאים לשימוש ממושך	20	17	5	18
ג. עלות הדרכת עובדים	25	21	9	17
ד. עלות תחזוקה	20	15	10	14
ה. הוצ׳ חד־פעמיות	20	18	11	14
	415	312	278	257

8. משא ומתן להתקשרות ולחתימת חוזה

לאחר בחירת הספק, יש לנהל משא ומתן להתקשרות עמו. החוזה עם הספק מבוסס בדרך־כלל על חוזה אחיד של הספק, אך עדיין נותרים סעיפים מסוימים למיקוח או להדגשת יתר. בשלב זה חשוב להגדיר באופן ברור וחד־משמעי מהי המערכת שעל הספק לספק ומהם המדדים שיקבעו את גמר ביצוע העבודה. כן חשוב להגדיר כיצד יטופל בתוספות ובשינויים שיידרשו בתוך ביצוע העבודה.

הבעיות של טיפול בשינויים ובתוספות, וכן קביעת גמר ביצוע העבודה, הן בעיות קריטיות כשמדובר במערכת מידע, מאחר שזו אינה מוצר מוחשי הניתן להגדרה סופית וחד־משמעית בשלב ההתקשרות; כמו־כן, מאחר

שמערכות מידע פועלות בארגונים חיים בעלי אופי דינמי — חלים שינויים הן בארגון והן בדרישות ממערכות המידע בתוך מהלך פיתוח המערכות. שינויים כאלה עלולים להשפיע על ההתקשרות עם הספק.

חשוב להדגיש בהסכם שנחתם גם את חלוקת האחריות בפיתוח ובייישום המערכת בין הארגון-הלקוח לבין הספק. יישום מערכת מידע בארגון איננו פעולה חד-צדדית המבוצעת על-ידי הספק בלבד, אלא בשילוב בינו לבין הארגון, תוך חלוקת משימות ואחריות.

בדרך-כלל ישתתף בשלב המשא ומתן להתקשרות ספק אחד; לעתים — יותר מאחד. קיימת אפשרות לרכוש גם מערכת מעורבת ממספר ספקים, אם יש יתרונות להתקשרות כזו. החיסרון העיקרי הוא בעומס המינהלי הנוצר כתוצאה מהצורך לבוא במגע עם מספר ספקים ובקשיי התיאום ביניהם.

התוצר של שלב זה — חוזה חתום לאספקת המערכת הנדרשת. לכאורה, מסתיים בזה תפקידה של קבוצת העבודה. למעשה, הסתיים שלב אחד — חשוב — במהלך יישום המערכת. השלבים העיקריים — ביצוע ויישום בארגון — עדיין נמצאים קדימה.

9. מעקב ביצוע

קבוצת העבודה חייבת להגדיר ברורות מהם הכלים שישמשו למעקב אחר ביצוע העבודה על-ידי הספק ועל-ידי גורמים שונים בארגון — בהתאם לחלוקה הנדרשת. בהיעדר מעקב ומדדים להתקדמות, קשה להניח שהעבודה תבוצע כראוי. המעקב נעשה במספר מישורים: על פני ציר הזמן, מבחינת המערכות המפותחות והמיושמות, לפי אישורי התאמה הניתנים על-ידי גורמים שונים בארגון וכדומה.

מעקב הביצוע חייב לקבל ביטוי בהסכם ההתקשרות עם הספק וכן בחלוקת האחריות לביצוע העבודה בין הספק והארגון.

הדגשים מיוחדים הקשורים בבחירת תוכנה

תוכנה ניתנת לפיתוח עצמי, לרכישה מיצרן, או לרכישה מספק אחר כלשהו (כגון ארגון אחר אשר פיתח תוכנה דומה). כאשר החליט הארגון לרכוש תוכנה, יש לנתח את התוכנות המוצעות, לבודקן ולהחליט על התוכנה הספציפית שתירכש.

כפי שהוזכר קודם לכן, קיימות שלוש קבוצות עיקריות של תוכנות: תוכנה בסיסית, תוכניות שרות ותוכנה יישומית. אנו נתרכז בעיקר בתוכנה היישומית.

כדי שתוכנה תתאים לדרישות הארגון, יש לבצע בדרך-כלל שינויים, הניתנים לעתים לביצוע בבית, אבל אפשר לדרשם גם מהמוכר.

ניתן לחלק את השינויים לשלוש קבוצות עיקריות: תיקונים של שגיאות קיימות, שינויי התאמה ושיפורי ביצועים.

מקורות לחבילות תוכנה

מספר המקורות הוא רב כמספר האפשרויות של היישומים השונים. בדרך־כלל ניתן לסווגם לקבוצות עיקריות:

— ספקי תוכנה מוכנה (turnkey suppliers)
— משווקי המחשב (חומרה)
— משתמשים אחרים (לעתים משווקת התוכנה ע״י יצרני המחשב)
— ספקים עצמאיים
— מוסדות להשכלה גבוהה או מוסדות אחרים ללא מטרת רווח.

מה לבחון?

עלויות: כדאי לבחון את עלות־התוכנה החדשה מבחינת עלות/תועלת.
קיום התוכנה: כמה זמן לאחר ההחלטה תהיה התוכנה מוכנה ליישום? מהם הקשיים הצפויים בהתקנת התוכנה?
אחזקה: מי יהיה אחראי על אחזקת התוכנה ועל ביצוע השינויים במידת הצורך?
התאמת התוכנה: האם התוכנה תתאים לדרישות הספציפיות של הארגון?
שינויים: האם ניתן לבצע שינויים בקלות? מי יבצע?
תיעוד: האם קיים תיעוד ובאיזו רמה?
שימוש במשאבים: מהם המשאבים הדרושים להפעלת התוכנה?

בתהליך הבחינה יש לבחון בזהירות כל אחת מהנקודות המצויינות לעיל. ניתן להשתמש בשיטת הציונים, כפי שהוצג בתהליך בחירת החומרה.
בחירת תוכנה היא לעתים תהליך קשה ומסובך יותר מבחירת חומרה. יש לזכור כי הדבר החשוב ביותר בתהליך זה הוא בדיקת התאמת התוכנה לצורכי הארגון. בחינה זאת ניתנת לביצוע במתקן הספק, אבל עדיף לעשותה במתקן הארגון, דבר המאפשר שיתוף רב יותר של בעלי תפקידים בבחינת התאמת התוכנה.

הערכת יישומי מערכת המידע

ניתן לבדוק את יישומי מערכת המידע בשלושה מדדים: טכני, תפעולי, וכלכלי. עבור יישומים חדשים מהווים המדדים הללו מדדי כדאיות, עבור היישומים הקיימים הם מהווים מדדי ביצוע:

1. בדיקות כדאיות/ביצוע טכני — האם ניתן לבצע את היישום בטכנולוגיה הקיימת?

2. בדיקת כדאיות/ביצוע תפעולי — האם המערכת פועלת (תפעל) בהצלחה? האם המשתמשים ישתמשו בה?
3. בדיקת כדאיות/ביצוע כלכלי — באיזו מידה רבה התועלת מהעלות של המערכת?

בדיקת כדאיות טכנית

בדיקת כדאיות טכנית בוחנת אם השיטה המוצעת לעיבוד נתונים אפשרית. יישומים רבים אינם ניתנים לביצוע בציוד ובתוכנה העומדים לרשות הארגון. במסגרת בדיקה זאת נשאלות, בין השאר, השאלות האלה:

— האם מספיקה מהירות העברת המידע?
— האם יש זיכרון חיצוני מספיק לאחסון הנתונים?
— האם יחידת העיבוד המרכזית יכולה לענות על הדרישות בזמן תגובה סביר?
— האם קיימות שיטות לפתרון כל הבעיות?
— האם מערכת ההפעלה מותאמת לגישה התפעולית?

לאחר הפעלת יישומים מסוימים, יתכן שהערכה מחדש תגלה חוסר יעילות הנובע מאי־יכולת הציוד והתוכנה לסייע כנדרש ליישומים הללו. יתכן שפעולה תעובד בזמן אמתי, אך באטיות רבה, משום שלמחשב הנדון אין יכולת לעמוד בעומס.

בדיקת כדאיות תפעולית

שיקולי בדיקת הכדאיות התפעולית מתייחסים לשאלות:
האם ניתן לספק את נתוני הקלט? האם ניתן להשתמש בפלט? לדוגמה, קיימת אפשרות טכנית שסוכני המכירות ידווחו על כל מכירה באמצעות טלפון, אך מבחינה תפעולית אין זה מעשי. קיימת אפשרות טכנית להדפיס דוחות מחשב של 100 עמודים, אך אין אפשרות תפעולית יעילה לנצלם. ניתן למקם מסוף במשרדו של כל מנהל, אך הניסיון מראה כי השימוש בו מצומצם ביותר.
הערכה של יישומים לאחר הפעלתם בוחנת בעיקר את מידת יעילותם, את אחוז השגויים וכד׳. מחקרים הראו כי חשוב להעריך בפרקי זמן קבועים את מידת היעילות של דוחות מחשב. קיימת נטייה להמשיך בפרסום דוח קיים, אף שאינו נמצא בשימוש שוטף, בגלל הסברה כי יועיל בעתיד. השיטות לגילוי תופעה זו הן:

— הפסקת פרסום הדוח כדי לראות אם יבקשוהו לכשלא יגיע
— סקירה תקופתית של הדוחות המופצים
— העמסת עלויות, אשר תניע מנהלים לצמצם ככל האפשר את פרסום הדוחות.

בדיקת כדאיות כלכלית

בעת בחינת ההצעה יש לערוך בדיקת כדאיות. לאחר התקנת המערכת יש להעריך מדי פעם את היחס של עלות/תועלת (cost/benefit).
נוח לחלק את העלויות והתועלות של מערכת עיבוד נתונים כך:

— עלויות:
א. נמדדות או נאמדות עם שונות נמוכה
ב. נאמדות עם שונות בינונית או גבוהה.

— תועלות:
א. נמדדות או נאמדות עם שונות נמוכה
ב. נאמדות עם שונות בינונית או גבוהה.

דוגמאות ניתנות בטבלה 5.5.

הוצאות ההפעלה השוטפות של מערכת ידועות כמעט בוודאות, ולכן נכללות הן בקבוצת העלויות הנאמדות עם שונות נמוכה. אמידת העלויות שיש להעמיס על יישום בודד קשה ביותר, משום שחלק מההוצאות משותפות ליישומים אחרים. ניתן לאמוד את העלויות הקשורות לפיתוח יישומים, אך שונותן נעה בדרך כלל בין 25% ל־100%. ישנן מספר עלויות משמעותיות, שנוהגים לזונחן משום ששונות אומדניהן גבוהה מאוד והן משתייכות למחלקות ארגוניות אחרות יותר מלעיבוד נתונים. לדוגמה:

עלויות הרצת המערכת	— העלויות הקשורות בהכנסת המערכת לשימוש הן משמעותיות לעתים, אך אינן מדווחות: הדרכת משתמשים בזמן עבודתם הרגילה, וכן ירידת יעילותם במשך תקופת ההדרכה.
התנהגות עובדים	— התנגדותם או אי־שביעות רצונם של העובדים ביחס למערכת החדשה יכולה לגרום לירידת יעילות עבודתם.
טיפול בשגויים	— עלות הטיפול בשגויים — בדיקה, תיקון והרצה מחדש — משמעותית, אך קשה לאמידה משום שהיא מפוזרת בארגון.
אבדן הכנסות כתוצאה משגיאות בעיבוד	— אי־שביעות רצונם של הלקוחות, הנובעת מרמת שגיאות גבוהה, קשה לאמידה, אך קיימת.

טבלה 5.5: דוגמה לעלויות ותועלות

	עלויות	תועלות
שונות נמוכה	ציוד, תוכנה נקנית, התקנת המערכת, כוח־אדם — מפעילים, אספקה — נייר וכד', אחזקה	חסכונות הנובעים מהקטנת הוצאות אחרות
שונות בינונית	פיתוח יישומים	חסכונות הנובעים מצמצומים בנכסים כגון מלאי, שיפורים בניהול נכסים, גידול בתשואות, צמצום בהלוואות.
שונות גבוהה	"הרצת" המערכת, הקטנת יעילות כתוצאה מהתנגדות עובדים, טיפול בשגויים, אבדן פריון כתוצאה משגיאות	גידול בפריון כתוצאה מצמצום זמן שירות והרחבת אפשרויות גישה למידע, צמצום עלויות כתוצאה משיפור דיוק, ביצועים טובים יותר הנובעים ממידע מעודכן יותר והרחבת האמצעים לקבלת החלטות.

שיטות מימון המחשב והתוכנה

המסחר במחשבים החל בהשכרת ציוד. רק לאחר 1965 התחילה חברת י.ב.מ. למכור ציוד מחשבים. עד היום מעודדות חברות מחשבים השכרת ציוד, דבר המעודד מכירות, שכן הלקוח אינו צריך להשקיע מיד השקעה גדולה.
למעלה משני שלישים מהמחשבים המצויים כיום בעולם שכורים; בישראל כמעט כל המחשבים הבינוניים והגדולים הם שכורים. מספר סיבות לכך והעיקרית שבהן היא בעיית ההתיישנות הטכנולוגית של הציוד. על־ידי השכרת המחשב ניתן להחליפו בכל זמן, דבר שהוא קשה ביותר כאשר המחשב קנוי. אולם כפי שיתואר להלן, נראה כי בכל זאת האופציה הטובה ביותר עשויה להיות רכישת המחשב ולא שכירתו, או אולי שכירתו מיד שלישית, כמקובל בארה"ב ובאירופה.*

* בשכירות מיד שלישית רוכש גוף מסוים את המחשב מיצרן המחשבים ומשכירו במחירים זולים ממחירי היצרן. שיטה זאת מבוססת על העובדה שמשך החיים של המחשב ארוך יותר ממה שמניחים יצרני המחשבים.

הבסיס להחלטה צריך להיות כלכלי ומבוסס על הערך הנוכחי המינימלי של תזרים המזומנים הרלוונטי לאחר מס. למעשה, בחירת שיטת המימון של המחשב דומה לבחירת שיטת המימון של כל ציוד אחר הנרכש על־ידי הארגון.

קיימים כמובן גורמים אחרים אשר צריכים להילקח בחשבון:

מיקוח: אם הציוד שכור קיימת האפשרות להפעיל לחץ על היצרן כדי שיספק שירותים טובים יותר; במקרה שהציוד קנוי — אפשרות זו פחותה. רוב חברות המחשבים מעוניינות בהשכרת ציוד יותר מבמכירתו, לפיכך מושפע בדרך־כלל השירות הניתן מצורת ההתקשרות עם היצרן.

התיישנות: ההצדקה העיקרית לשכירת מחשב היא התיישנותם של מחשבים. כאשר שוכרים ציוד ניתן להחליף את המחשב בהופיע ציוד חדש. בישראל שונה הדבר במידה מסוימת עקב שיטת תשלומי המכס המאפשרת החלפת ציוד רק בזמנים קבועים לפי תשלומים הנקבעים על־ידי המכס. אולם יש להביא בחשבון כי החלפה תדירה אינה רצויה, אם ההשקעה בתכנון המערכת (כולל ניתוח, תיכון ותכנות) היתה גדולה וטרם כוסתה.

שלושה חלקים להתיישנות:

התיישנות פיזית: כאשר המחשב מתיישן מבחינה פיזית יש להחליפו. אולם רוב המחשבים בנויים כך, שאורך חייהם יכסה את מחיר הקנייה. אורך החיים הפיזי של מערכות מחשבים תלוי במרכיבים והוא נע בין 8 ל־14 שנים. התיישנות פיזית אינה צריכה אפוא להילקח בחשבון, אם נקודת האיזון בין האופציות השונות היא פחות מ־8 שנים.

התיישנות טכנולוגית: תעשיית המחשבים עברה זעזועים טכנולוגיים לאורך ההיסטוריה הקצרה יחסית שלה — כ־25 שנה. השינויים במחשבים מדור לדור שיפרו את יעילותם ואת ביצועיהם. אולם הואיל והיצרנים השקיעו כספים ומאמצים בפיתוח מערכות חדשות, יש להניח שעד שלא יכוסו כל ההשקעות לא יהיה כדאי להם לצאת עם מערכות חדשות. לפיכך, מרחק הזמן בין דור לדור הוא כ־6 שנים לפחות. תופעה זאת אופיינית במיוחד לשוק המחשבים. נוסף לכך, קצב השינויים הטכנולוגיים קטן עם הזמן. נראה שקיים גבול לשיפורים שאפשר להשיג. לכן עניין ההתיישנות הטכנולוגית איבד מערכו בשנים האחרונות, ובמקרה של קנייה הסיכון הוא קטן ממה שנראה על פני השטח.

חשוב ביותר להדגיש כי שינויים טכנולוגיים אינם צריכים בהכרח להשפיע על הלקוח כל עוד המערכת שברשותו עונה לדרישות והמצב האובייקטיבי אינו מחייב התרחבות או שינוי.

התיישנות יישומית: הגורם האחרון שצריך להילקח בחשבון הוא ההתיישנות היישומית, לאמור — נקודת הזמן שבה לא יכולה עוד מערכת המחשב לענות לדרישות הלקוח בצורה יעילה. נקודת זמן זאת נקבעת על־ידי ההשקעה בתכנון יותר מאשר על־ידי מחיר הציוד. בדרך־כלל, מערכת צריכה להיות מתוכננת

בצורה שתאפשר כיסוי כל הוצאות התכנון (ניתוח ותיכון, תכנות, בדיקות וכד׳). ברוב הארגונים והחברות עולה עלות הפיתוח והתיכון של מערכות על מחיר הציוד, לפיכך עובר אורך החיים הכלכלי של המערכות את נקודת האיזון שבין שכירות לקנייה.

גורמים כלכליים

המרכיבים העיקריים של העלויות באופציות השונות הם כדלהלן:

שכירות:

— תשלומי שכירות חודשיים
— תשלומים נוספים עבור שימוש מעל משמרת אחת (בדרך כלל משמרת אחת = 176 שעות לחודש)
— תשלומים עבור מכס, מס־קנייה מס־שירותים.

שכירות יד שלישית:

— תשלומי שכירות חודשיים (בלי שעות נוספות)
— אחזקה (בדרך־כלל בתוך תשלומי השכירות)
— מכס, מס־קניה מס־שירותים.

קנייה:

— השקעה ראשונית בקניית הציוד
— אחזקה (חוץ מתקופת האחריות, שהיא בדרך־כלל כשלושה עד שישה חדשים)
— ביטוח
— אפשרות שימוש בפחת מואץ לצורכי מס־הכנסה.

קנייה בתשלומים:

— קיימת אפשרות לקנות את הציוד בתשלומים
— יתר המרכיבים כמו בקנייה.

יתר המרכיבים הכלכליים, כגון עלות הבניין, מיסוי רגיל וכד׳, שווים בכל האופציות. לשם ניתוח והשוואת האלטרנטיבות השונות, מקובל להשתמש בשיטת הערך הנוכחי של זרם המזומנים. בשיטה זאת ניתן להשוות את החלופות השונות ולהגדיר את נקודת הזמן אשר לאחריה רכישת מחשב כדאית יותר, כלומר אם זמן השימוש עובר נקודה זאת, נרכוש את המחשב.

פרק 6.

היבטים ארגוניים וניהוליים של מערכות מידע ממוחשבות

ארגון ומבנה יחידת מערכות מידע

מבנה יחידת עיבוד נתונים מבוסס על עקרונות ארגוניים המשותפים ליחידות ארגוניות רבות, אך על עקרונות אלו נוספים העקרונות המיוחדים לארגון פעולת עיבוד הנתונים והנובעים מאופייה המיוחד של פעילות זו.

יחידת עיבוד נתונים צריכה, ככל ארגון, לקיים את המטלות האלה:

— להכיר את הסביבה שבה פועלת היחידה ואת מטרותיה.
— הגדרה ברורה של יחסי סמכות ואחריות.
— הגדרה ברורה של קווי דיווח בתוך היחידה ומחוץ לה.
— הגדרת פונקציות המביאה לניצול כלכלי יעיל של משאבים.

למעשה מספקת יחידת עיבוד הנתונים שירות לשאר חלקי הארגון. לעתים הציוד, המתכננים והמתכנתים הם חלק מהמערכות השונות בארגון, לפיכך אחת המטרות של מערכת עיבוד נתונים טובה היא להביא לשילוב של פעולות עיבוד הנתונים בין חלקיו השונים של הארגון. תכונות ייחודיות אלו גורמות להבלטת מספר תפיסות:

— יש למקם את הפעילות לעיבוד הנתונים כך שתוכל לספק את צורכיהם של כל חלקי הארגון ביעילות.
— המבנה הארגוני צריך להיות מותאם לצורך בשינוי ובעדכונים מתמידים הנובעים מההתפתחות המהירה בענף המחשבים.
— על המבנה הארגוני להבליט את פונקציית התכנון, התזמון והפיקוח כדי שהיחידה לעיבוד נתונים תמלא את תפקידה.

— פונקציית עיבוד הנתונים צריכה להיות מאורגנת כך, שלא יהיו לשום פרט או ליחידה קטנה שליטה מלאה על המידע המעובד. האחריות צריכה להתחלק כך, שמאמצי קבוצה אחת יבדקו את פעילותה של קבוצה אחרת.

גורמים סביבתיים

הבנת סביבתה של הפעילות לעיבוד הנתונים חשובה ביותר בעת ארגון הפונקציות השונות. בתהליך הארגון יש לפתור שתי שאלות עיקריות:

— כיצד מתקשרת פונקציית עיבוד הנתונים עם הארגון כולו?
— מהן המטרות הייחודיות שעל פונקציית עיבוד הנתונים לשרת?

כדי לענות על שאלות אלו על המארגן לקבל תמונה ביחס לנתונים האלה:

— מהו המצב של עיבוד הנתונים בארגון, מהן הדרישות הנוכחיות ומהן תכניות הפיתוח לעתיד.
— מהי מדיניות ההנהלה הבכירה ביחס למטרות עיבוד הנתונים.
— כיצד משרתת יחידת עיבוד הנתונים את המשתמשים ומהי רמת השירות.

מטרות

הסיבה העיקרית לקיום פעילות עיבוד הנתונים בארגון — השגת מטרות הארגון. אחדות מהמטרות העיקריות מנויות להלן:

— תמיכה בפרוייקט פיתוח אחד או יותר (למשל תכנון מוצרים, פיתוח תוכנה וכד׳).
— תמיכה בפעילויות תפעול (כגון פיקוח על הייצור, ניהול חשבונות עובר ושב בבנקים וכד׳).
— תמיכה בהנהלה (למשל מערכות מידע ניהולי, ניהול בקרה ותקציב וכד׳).
— מתן שירותי מחשב למשתמשים במידת הצורך.

כמובן שפעילות מסוימת יכולה לשרת יותר ממטרה אחת. אולם המטרה העיקרית היא זו המשפיעה על עיקר המבנה הפנימי של הפעילות.

גורמים אחרים

גורמים סביבתיים אחרים:

— מספר המשתמשים בעיבוד נתונים וערכם: האם קיים מספר גדול של משתמשים קטנים, או מספר קטן של משתמשים גדולים? כיצד מתחלק עומס העבודה ביניהם? וכד׳.
— תלות ההנהלה בשירותים הניתנים על־ידי עיבוד הנתונים: האם פלט עיבוד הנתונים והקבצים הם דבר חשוב ונדרש להנהלה לתכנון? האם דרושה להנהלה גישה מהירה למידע?

— תלות החלק התפעולי בשירותים הניתנים על־ידי עיבוד הנתונים: האם עיבוד הנתונים הוא חלק אינטגרלי של העבודה המבוצעת? האם זמינות מהירה של המידע היא גורם חשוב בתפעול השוטף?
— סוג עיבוד הנתונים: האם הוא מבוסס בעיקר על מסדי נתונים? עיבודים מדעיים? פיקוח על תהליכים? וכד׳.
— איזה חלק מהתכנון והתכנות נעשה בידי המשתמשים עצמם? האם פונקציית עיבוד הנתונים פועלת כלשכת שירות? האם מקיימת פונקציית עיבוד הנתונים מאגר של אנשי מקצוע? האם כל עבודת הפיתוח נעשית על־ידי פונקציית עיבוד הנתונים?
— היחס שבין עבודות פיתוח לבין עבודות אחזקה: האם מרבית המשאבים מוקדשים לעבודות פיתוח או לעבודות אחזקה ולעדכונים?
— תחזיות של עומס עבודה ולוח זמנים: האם ניתן לתזמן בקלות עבודות עיבוד נתונים? האם ניתן לבצע את העבודות בזמן־סבב קצר ללא הודעה מראש?
— סוגי הקלט למערכות עיבוד הנתונים: בעזרת מסופים או כרטיסים? האם יש צורך להפוך את נתוני הקלט למצב שבו ניתן לקלטם? וכד׳.

סביבות טיפוסיות

מספר הצורות בארגון הסביבתי הוא כמספר הקומבינציות של גורמי הסביבה. בחרנו בארבע דוגמאות כדי להמחיש כיצד פרופילים סביבתיים שונים זה מזה.

הפעילות הטיפוסית הראשונה אופיינית בדרך־כלל ליחידת עיבוד נתונים בגודל קטן עד בינוני, המשרתת את ההנהלה ואת המדורים כספים, משכורת, רכש וניהול מלאי. הקבצים מעודכנים בזמנים קבועים, וכן מיוצר פלט בזמנים קבועים. רוב הנתונים מיוצרים מטופסי מקור והמשתמשים תלויים לגמרי בפעילות היחידה. פיתוח המערכות נעשה כבר ברובו, ופעילות המתכננים והמתכנתים קשורה בעיקר באחזקה ובתיקונים קטנים.

הפעילות הטיפוסית השנייה עיקרה פיתוח. בעניין זה הדרישה לפיתוח ותכנות מערכות גדולה ביותר והדבר נעשה בדרך־כלל בשיתוף עם המשתמשים. עומס העבודה הוא גדול למדי ובלתי צפוי. מספר משתמשים עומדים על כך שיקבלו עדיפות שווה וזמן תגובה מיידי. בדרך־כלל יהיה המחשב בינוני עד גדול והקלט יגיע בצורות שונות.

הפעילות הטיפוסית השלישית היא זו הנותנת שירותי מחשב, אם על־ידי שירותים פנימיים ואם על־ידי לשכת שירות מסחרית. מטרתה העיקרית לענות על דרישת המשתמשים למספר גדול של יישומים ומטרות. קיים מספר גדול של משתמשים, חלק מהם קשור למחשב על־ידי מסופים והחלק האחר מעבד את יישומיו בעיבוד במכלול. ניתן לצפות את עומס העבודה אך לא על בסיס של עבודות ספציפיות.

המשתמשים מבצעים את כל התכנון והתכנות בעזרת אנשי יחידת עיבוד הנתונים, כדי להגיע לשימוש יעיל יותר של הציוד.

הפעילות הרביעית היא זו המשתמשת בקשר ישיר למסדי הנתונים. מסדי הנתונים הם החלק העיקרי והחשוב של הפעילות לעיבוד הנתונים, ולמעשה חלק אינטגרלי מפעילות הארגון. מסדי הנתונים הם גדולים, ומעודכנים באופן שוטף על־ידי קשר עם מסופים המספקים את רוב נתוני הקלט. שאילתות רבות מוזרמות למסדי הנתונים וניתן לחזות בדרך־כלל את העומס הצפוי.

יישומים חדשים מתווספים ללא הרף כאשר פיתוח המערכות נעשה בצורה מרכזית על־ידי יחידת עיבוד הנתונים. במקביל נעשית אחזקה שוטפת של המערכות הקיימות, ולעתים מבוצעים שינויים קלים במערכות אלו.

מיקום יחידת עיבוד הנתונים במסגרת הארגון

קיימות שלוש אלטרנטיבות למיקום יחידת עיבוד הנתונים במסגרת הארגון:

מיקום תפעולי

לפי תרשים 6.1 שלוש פעילויות של עיבוד נתונים מתקיימות באופן עצמאי במסגרת הארגון ומשרתות את פונקציית המימון, ההנדסה והייצור. שלושת הפונקציות המרכזיות האחרות, מחקר ופיתוח, שיווק ומינהל יכולות לשכור שירותים עבור אחת או יותר מהפעילויות בארגון או מחוצה לו.
מבנה כזה מתאים במיוחד לארגון מבוזר, ארגון שאין בו בדרך־כלל שילוב של מסדי נתונים. מבנה כזה יתכן גם כתוצאה מפיזור גיאוגרפי, או מקיום פונקציות שונות לחלוטין באגפים השונים. העברת מידע בין הפעילויות העצמאיות תיתכן באמצעות מערכת תקשורת, העברת סרטים, או העברה ידנית.
יתרונות המבנה:

א. פיקוח צמוד של הפונקציות הארגוניות על יחידת עיבוד הנתונים שבשליטתם.

ב. כל יחידה יכולה לרכוש אך ורק את הציוד הדרוש לה, מבלי להיזדקק ליחידה גדולה הדורשת התאמות לעבודות שונות.

ג. מתאים לגישה של עלות ורווח לפי מרכזי רווח.

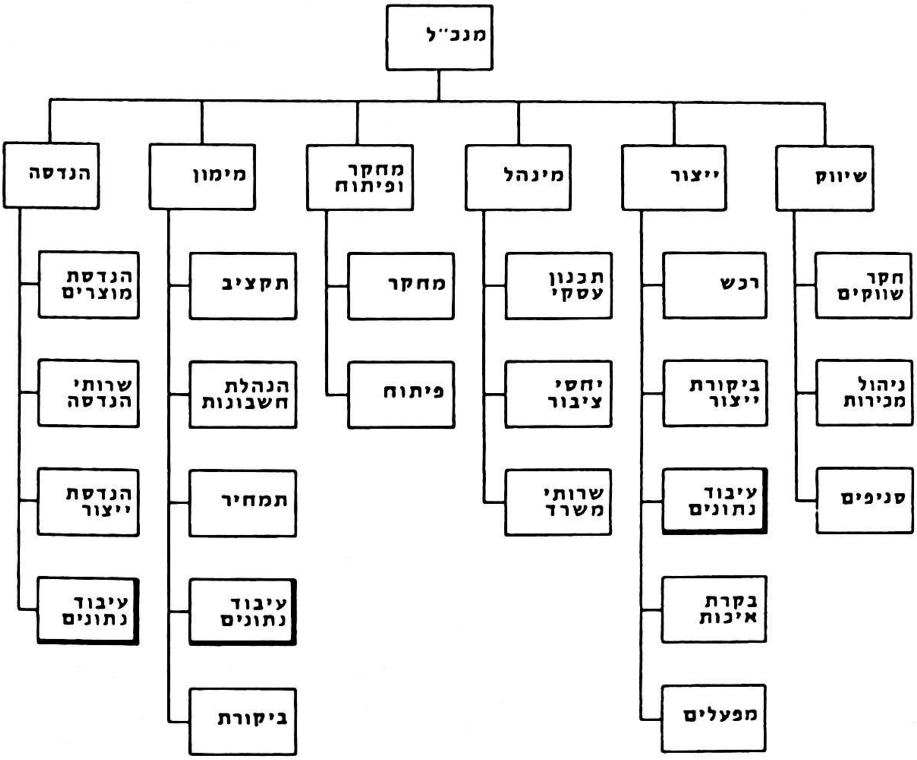
חסרונות המבנה:

א. יתכן בזבוז של ציוד ושל כוח אדם עקב כפילות בתפקידים.

ב. קיימים קשיים בשילוב של מסדי נתונים.

ג. קשה לקבוע תקנים ולשלוט על תהליך עיבוד הנתונים והפיתוח, משום

שהיחידות כפופות לפונקציות ארגוניות ולא למרכז לעיבוד נתונים.
אופנים שונים לשיטה זו: המשותף להן הוא בדרך־כלל הניסיון לקשר את פעילויות עיבוד הנתונים העצמאיות. באמצעות תפקיד "החותך" את הקווים הארגוניים בתרשים.



תרשים 6.1: תרשים ארגוני — עיבוד נתונים הוא חלק מהיחידות התפעוליות

מיקום במסגרת אגף המינהל

בתרשים 6.2 עומדת יחידת עיבוד הנתונים כמחלקה שוות־מעמד באגף המינהל, המספק שירותים מרוכזים לכל יחידות הארגון. אם יש צורך במיקום הציוד והמשאבים במקומות מרוחקים (מפוזרים מבחינה גיאוגרפית), או בפונקציות שונות בארגון, נשאר הפיקוח והניהול בידי יחידת עיבוד הנתונים המרכזית.

בגישה זו "משלמות" המחלקות השונות עבור שירותי המחשב לפי קביעת מחירים פנימית של הארגון. היא מותאמת לארגונים בינוניים וקטנים וכן לארגונים גדולים שקיים בהם צורך בשילוב רב של מסדי הנתונים.

יתרונות הגישה:

א. כל המחלקות מקבלות טיפול שווה.

ב. כל המשתמשים יכולים ליהנות מיכולת חישובית גדולה, עקב הימצאות מחשב גדול.

ג. שליטה נוחה על מחלקת עיבוד הנתונים ובתוכה.

ד. אפשרות לחסוך בכוח־אדם, בעיקר בדרגי הנהלת היחידה.

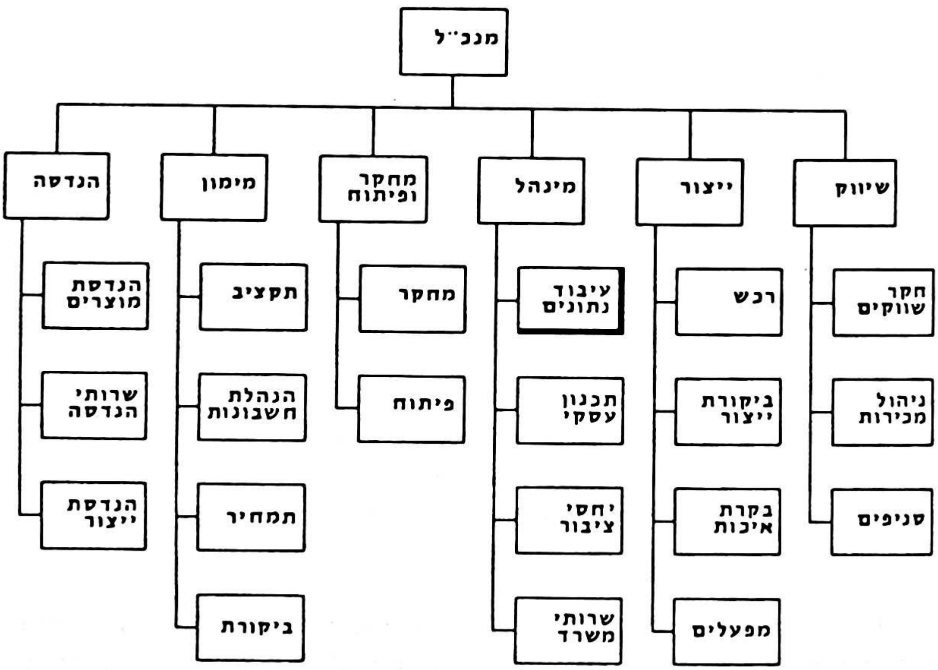
ה. אפשרות הרחבת פעילותו של משתמש במסגרת היחידה ואין צורך בהרחבת ציוד לשם כך.

חסרונות הגישה:

א. המתכננים והמתכנתים אינם מכירים מקרוב את הפונקציות השונות בארגון.

ב. יש צורך בתמחיר מדויק של שירותי המחשב, ולא, תיווצר נטייה "לבזבז" את שירותי המחשב.

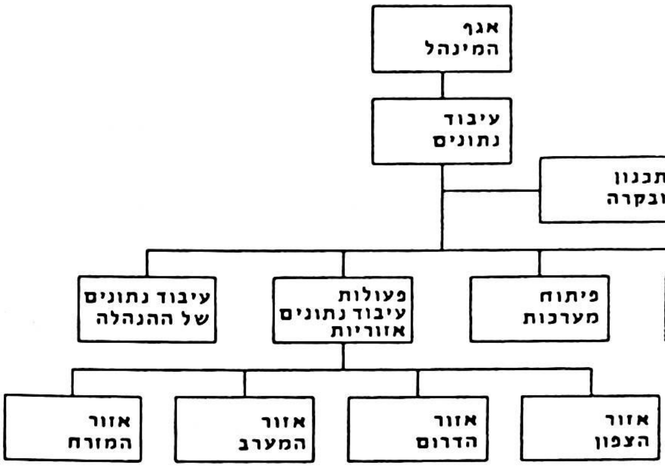
ג. קיים קושי בקביעת עדיפויות והקצאות בין המשתמשים השונים.



תרשים 6.2: תרשים ארגוני — עיבוד נתונים הוא חלק משירותי אגף המינהל

תרשים 6.3 מציג צורה אחת של גישה זאת אשר מתגברת על חלק מהחסרונות שציינו לעיל. במקרה המוצג מפעילה מחלקת עיבוד הנתונים חמישה מרכזי חישוב: ארבעה במשרדים אזוריים ואחד במשרד הראשי של הארגון. המרכזים האזוריים משרתים את המשרדים והמפעלים באזור הגיאוגרפי שבאחריותם, ואילו המרכז במשרד הראשי משרת את כל הפעילויות המרוכזות במשרד זה וכן את היישומים אשר משתמשים בנתונים אשר נאספו ברמה מקומית ולא עובדו שם. כוח האדם במשרדים האזוריים הוא חלק אינטגרלי ממחלקת עיבוד הנתונים המרכזית וכל משרד אזורי כפוף בנושאי עיבוד הנתונים למרכז. פיתוח ותכנות יישומים סטנדרטיים נעשים במרכז, אולם יתכן שמנתחי מערכות ומתכנתים יימצאו במשרדים האזוריים כדי לבצע עבודות הקשורות אך ורק למשרד האזורי.

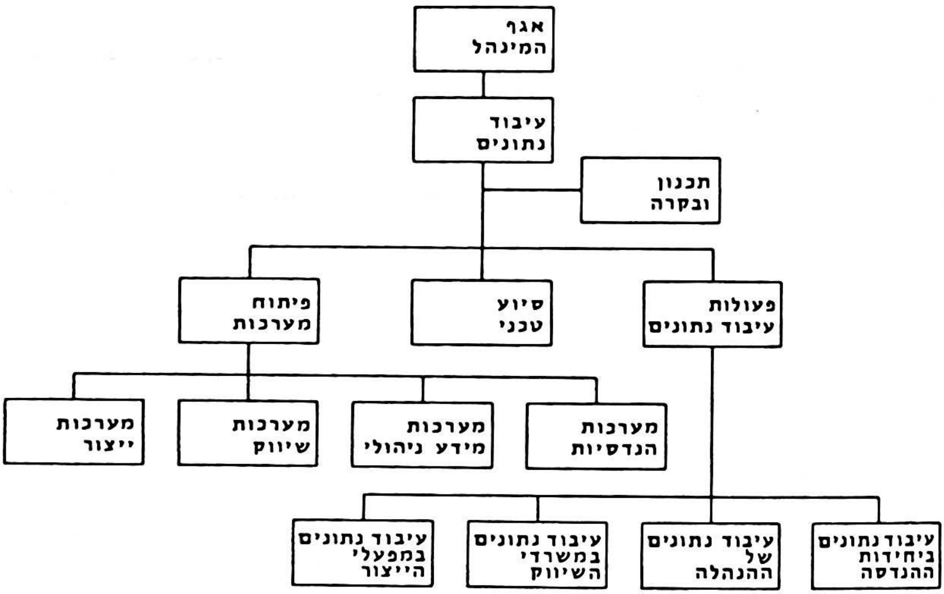
הרכב הציוד הוא שונה ומותאם לכל מרכז אזורי לפי צרכיו. כן שונה גם הציוד הנמצא במרכז. ניתן כמובן לקשור את המרכזים השונים למטה במרכז על־ידי שימוש במערכות תקשורת.



תרשים 6.3: יחידות עיבוד נתונים אזורית
כפופות למחלקת עיבוד נתונים מרכזית

תרשים 6.4 מציג צורה אחרת לגישה הנ״ל, אולם במקרה זה נמצאות תת־מחלקות של עיבוד הנתונים במחלקות התפעוליות עצמן. פיתוח המערכות נעשה במקרה זה על־ידי התמחות בארבעה יישומים עיקריים: ייצור, שיווק, ארגון והנדסה. קבוצות אלו פועלות על מערכות סטנדרטיות לשימוש באמצעות מחלקת

התפעול. גם במקרה זה פועל כוח האדם ברמה המקומית אולם הפיקוח והניהול נעשים על־ידי מחלקת עיבוד הנתונים המרכזית. גישה זו מאפשרת התמחות בשטחים שונים, אך משאירה את הפיקוח והניהול בידי מחלקת עיבוד הנתונים המרכזית.



תרשים 6.4 : יחידות עיבוד נתונים כפופות למחלקות התפעוליות

מיקום עצמאי

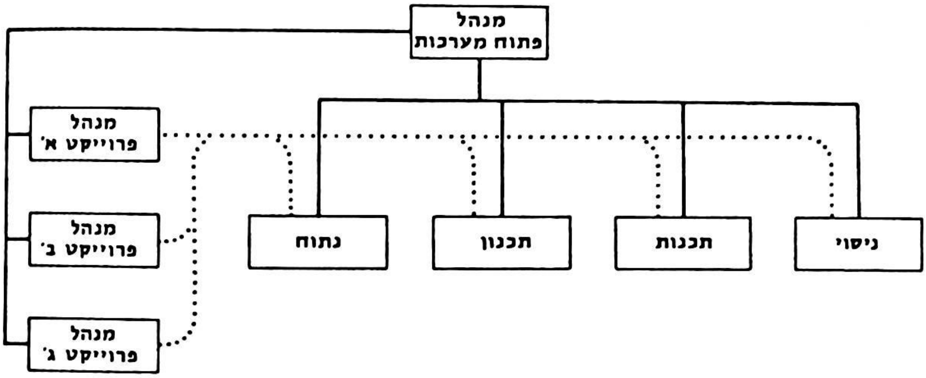
בגישה זו נמצאת מחלקת עיבוד הנתונים תחת פיקוח ישיר של מנהל הארגון כולו. למחלקה זאת ניתנים אותה חשיבות, עצמאות ותשומת־לב ההנהלה, כלמחלקות עיקריות אחרות בארגון (כגון לשיווק, למימון).

גישה זו דומה לגישה הקודמת שבה ממוקמת מחלקת עיבוד הנתונים במסגרת אגף המינהל, ותיתכן כפיפות באותן צורות של יחידות עיבוד נתונים באיזורים גיאוגרפיים שונים, או במחלקות תפעוליות שונות.

מבנה מטריצה

מבנה ארגוני זה מבוסס על מבנה פונקציונלי כמופיע בתרשים 6.4 בתוספת מנהלת פרוייקט עבור כל פיתוח נוסף. מנהל הפרוייקט אחראי על פיקוח ועל מעקב אחר תהליך הפיתוח. דוגמה למבנה זה מוצגת בתרשים 6.5. חשוב להזכיר

כי מבנה זה הוא ייחודי להתארגנות לפיתוח מערכות מידע. מסגרת זאת היא דינמית ומשתנה בהתאם לצורכי הארגון.



תרשים 6.5 : מבנה מטריצה

מבנה של יחידת מערכות מידע מורכבת

במקרה של יחידה גדולה יש לקחת בחשבון את כל הנאמר לעיל, על מנת לאפשר ניהול תקין וקל של פונקציות מערכות המידע בארגון. דוגמה למבנה כזה מוצגת בתרשים 6.6. היחידה מורכבת מחמישה מרכיבים עיקריים:

יחידת הניהול: מבצעת את כל הדרוש לניהול ואת כל האדמיניסטרציה הדרושה עבור היחידות האחרות.

יחידת הייצור: אחראית לכל עבודת הייצור, כגון הפעלת החומר, קליטה ופיקוח על הנתונים, טיפול בספריית הסרטים והתקליטים.

היחידה הכלכלית: אחראית על כל הצד הפיננסי של מערכות המידע, כגון תקציבים, תמחיר והנהלת חשבונות.

יחידת שירותי מידע: אחראית על פיתוח יישומים, אחזקה ועדכון עבור המשתמשים. כל יחידה מעסיקה מנתח מערכות, מתכנן מערכות ותוכניתן אחזקה; תוכניתני היישומים מושאלים ממאגר התוכניתנים. תקנים ונהלים מתואמים על-ידי מנתח המערכות הראשי.

יחידת משאבי פנים: אחראית לכל המשאבים הפנימיים, כולל מדידה והערכה של ביצועי מערכת המחשב, תכנון קיבולת, פיתוח ויישום של מערכות הפעלה ותוכנות עיקריות אחרות (כגון תוכנת מסדי נתונים), תקשורת, ניהול מסדי נתונים, הכנת נוהלי סודיות וביטחון ויישומם. יחידה זאת כוללת את מאגר התכניתנים.

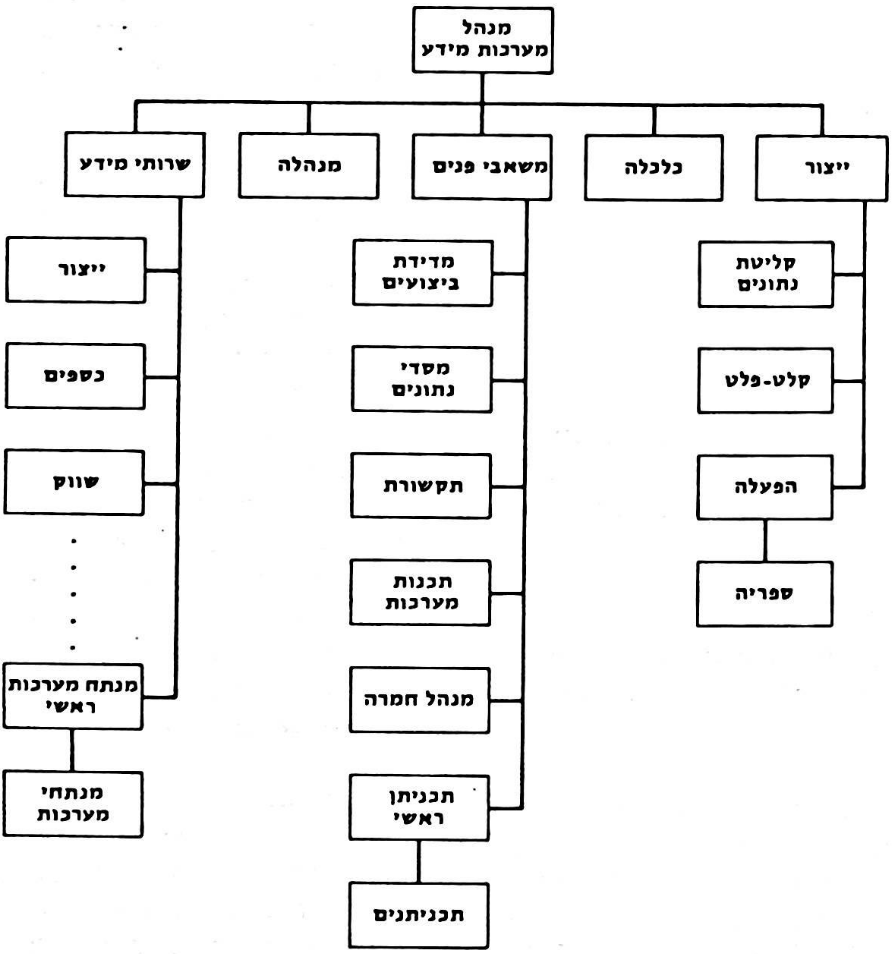
מבנה זה מבוסס על העקרונות האלה:

— פיתוח יישומים נעשה בידי אנשים המתמחים בנושא המסוים.

— המשתמשים אינם חייבים לבוא במגע עם מספר רב של בעלי תפקידים; קיים אדם אחד המתאם את דרישות המשתמש.

— הייצור מנוהל בצורה מרכזית.

— האנשים מחולקים לשתי קבוצות עיקריות: מקצוענים, אשר נותנים תמיכה טכנית ומנוהלים על־ידי מקצוען בכיר; אנשים אדמיניסטרטיביים, אשר נותנים את התמיכה הדרושה לניהול מחלקת מערכות המידע.



תרשים 6.6: מבנה של יחידת מערכות מידע מורכבת

פונקציות בעיבוד נתונים

בכל יחידת עיבוד נתונים צריכות להתבצע מספר פונקציות. צורת קיומן ותרגומן לתפקידים ספציפיים שונים מיחידה ליחידה, בהתאם לסביבה שבה היחידה פועלת.

פונקציות פיתוח מערכות

כל מערכת חייבת לעבור ניתוח, תיכון, תכנות וניסוי. כמו כן יש לתחזק אותה ולבוחנה באופן מתמיד. להלן הפירוט המתבקש:

1. ניתוח מערכות — מטרתה של פעילות זו לתת פתרונות טובים ביותר לצרכים נתונים, בהתחשב במטרות ההנהלה ובמשאבי הארגון (ציוד וכוח־אדם). בפעילות זו כלולים:
   א. בחירת המערכות שיש לפתחן.
   ב. ניתוח פעילויות קיימות ומתוכננות כדי לשפרן; איסוף מידע בקשר לפעילויות אלה וארגונו בצורה משמעותית לניתוח (תרשימי זרימה, גרפים וכו׳).
   ג. ניתוח עלות/תועלת של גישות אלטרנטיביות לתיכון המערכת.
   ד. ניתוח צורכי מידע ופיתוח מסדי נתונים המותאמים לצרכים.

2. תיכון מערכת פירושו — תיכון מדויק של המערכת על סמך הצרכים. הפלט של תיכון מערכות הוא מידע מדויק המאפשר למתכנת לבצע את תפקידו ללא צורך במידע נוסף. מידע זה כולל:
   א. מפרטים של מסדי הנתונים.
   ב. מרכיבי מערכת המחולקים לשלבים אופרטיביים, סדר שלבים מוגדר והקשר ביניהם (חוליות — תת־מערכות).
   ג. הגדרה מדויקת של צורכי עיבוד הנתונים בכל שלב, כולל ציוד ומערכת הפעלה (מפרט תכניות).
   ד. פירוט שיטות התכנות.
   ה. הגדרה הקובעת את דרכי הרישום של תהליכים ונהלים.
   ו. קביעת תהליכי הבדיקה והניסוי של תתי המערכת וקביעת הנהלים.
   ז. פירוט התהליכים שיתבצעו אם יהיה צורך לעדכן את המערכת בשלבי התכנות, הניסוי או היישום.

3. תכנות יישומים — תכנות השלבים השונים במערכת:
   א. תכנון לוגי.
   ב. קידוד ההוראות בשפה שנבחרה ליישום (תכנות).
   ג. תרגום (קומפילציה וניסוי התכנות).
   ד. תיעוד מדויק של התכניות וההוראות להפעלתן לפי הנהלים הקיימים ביחידה.

4. ניסוי המערכת:
א. תכנון שיטות ניסוי לבדיקת המערכת לפי התקנים שנקבעו.
ב. בחירת נתוני ניסוי המייצגים נכונה את התנאים שבהם תפעל המערכת.
ג. ניסוי המערכת, עדכונה ותיקון ליקויים.
ד. בחינת קבלה — קבלת המערכת, יישומה או דחייתה, והשלמת הפיתוח.

5. התקנת המערכת — הכנסת מערכת חדשה לפעולה:
א. תכנון שלבי המעבר למערכת החדשה.
ב. תכנון הוראות וחומר לימודי למשתמשים.
ג. הדרכת המשתמשים ואנשי עיבוד הנתונים במערכת החדשה.
ד. שילוב של המערכת החדשה עם המערכות הקיימות.

6. תחזוקת המערכת:
תיקונים ושינויים במערכת הנובעים מהפעלה שוטפת.
הפעולות הנדרשות הן למעשה תמצית הפעולות בסעיפים 1—5.

7. הערכת ביצועי המערכת:
מדי פעם בפעם יש לבדוק אם המערכת עומדת במטרות ובתקנים שנקבעו לה:
א. מבחינת התכונות האופרטיביות — זמני הפעלה, אחוזי שגיאה וכו׳.
ב. עלות.
ג. מידת ההתאמה למערכות אחרות ואפשרויות שילוב עם מערכות חדשות.
ד. מידת נצולת הציוד.
ה. מידת יעילות הנהלים.

פונקציות אופרטיביות

הפונקציות האופרטיביות בעיבוד נתונים אנלוגיות לכל מפעל יצרני אחר:

1. פיקוח ותזמון — מרבית יחידות עיבוד הנתונים עומדות לפני ביקוש משני סוגים:
א. מתוכנן.
ב. בלתי מתוכנן.
על היחידה לתזמן את העבודות ביעילות המרבית. עבודה זו כוללת:
א. תכנון עבור עבודות מתוכננות.
ב. פיקוח ותיאום יומיומיים, המביאים בחשבון קלקולים בציוד, עבודות בלתי צפויות וכו׳.
ג. פיקוח על זרימת נתוני הקלט והפצת הפלטים.
ד. קשר עם המשתמשים בנוגע לדרישותיהם ולמצב עבודותיהם.

2. הפעלת הציוד — כל הכרוך בהפעלה שוטפת של ציוד עיבוד הנתונים, הכולל:
   א. ציוד להכנת הקלט.
   ב. ציוד הפיקוח של מערכת התקשורת.
   ג. ציוד אחר, הכולל לדוגמה מכונות קלידה, מפרידות נייר.
   ד. הפעלה ופיקוח על ציוד המחשב, מסופים, ציוד היקפי המחובר ישירות למחשב, הפעלת מדפסות וכד׳.

3. פונקציית סיוע להפעלה:
   א. תחזוקה שוטפת של ספריית סרטים ותקליטים.
   ב. תחזוקה שוטפת של ספריית תיעוד הפעלה.
   ג. ניהול שוטף של מלאי חומרים הדרושים בתהליך.
   ד. איסוף נתונים ודיווח על ביצועי המערכת.
   ה. הגנה פיסית על הציוד.

סיוע טכני

1. פיתוח תקנים לעיבוד נתונים — קביעה, פרסום והשגחה על ביצוע כל הפונקציות לפי השיטות והתקנים שנקבעו. תקנים אלה קובעים:
   א. שיטת עבודה, אחריות וסמכות.
   ב. מהו הרישום הנדרש בכל פעולה.
   ג. מגבלות.
   ד. מהו מבנה הקבצים, גודל הקבצים, בטיחות הקבצים וכד׳.
   ה. ביצועים מינימליים הדרושים מכ״א וציוד.

2. ייעוץ טכני:
   א. ייעוץ להנהלה ולמשתמשים בתכנון ובניצול היכולת של יחידת עיבוד הנתונים.
   ב. ייעוץ לפתרון בעיות שוטפות ולשאלות של המשתמשים.
   ג. קיום ספרייה טכנית לשימוש המשתמשים.
   ד. פרסום עדכונים במערכת.
   ה. קיום קורסי שירות.

3. תכנות המערכת:
   א. בחירת מערכת הפעלה.
   ב. קביעת תקנים לשימוש בתוכנה כללית ובתכניות עזר.
   ג. סיוע וייעוץ למתכנתים בשימוש בתוכנה כללית ובתכניות עזר.

4. ניהול מסד נתונים:
א. תיכון מבנה מסד נתונים המותאם לצרכיה של היחידה והמתחשב גם בצרכים צפויים בעתיד.
ב. קביעת תקנים לשימוש במסד הנתונים.
ג. קביעת נוהלי בטיחות ביחס לשמירת מסד הנתונים.
ד. עדכון סעיפים א, ב, ג, בהתאם להתפתחויות שוטפות.
ה. אחזקת מילון למסד הנתונים.

5. ניהול מערכת תקשורת:
פיתוח, תכנון ופיקוח על מערכות תקשורת. פעילות זו כוללת:
א. קביעת התכונות הנדרשות.
ב. סינון האלטרנטיבות בהתאם לדרישות ולתכונות ורכישת הציוד והתוכנה המתאימים.
ג. ייעוץ והדרכת המשתמשים השונים בציוד ובנהלים.
ד. קביעת התקנים לשימוש ביחידות תקשורת.
ה. תחזוקה.

מינהלה והנהלה

הנהלת יחידת עיבוד נתונים כוללת את כל פונקציות הניהול הקיימות בכל ארגון העוסק בפעילות מורכבת ומגוונת. מודגשות במיוחד הפונקציות האלה:

1. פיקוח וניהול פרוייקטים: לעתים תמצאנה שתי שרשראות פיקוח ביחידת עיבוד נתונים: האחת השגורה בכל ארגון, והשנייה הכרוכה בניהול פרוייקט המאורגן לרוחב הקווים הארגוניים, דהיינו, לא במסגרת פונקצייה מסוימת בארגון.

2. תכנון: הערכתם של הדרישות העתידיות והמקורות הדרושים לסיפוק צרכים אלו. עקב ההתפתחות המהירה בענף המחשבים, יש צורך בהגדרה ברורה של טווחי התכנון: קצר, בינוני וארוך. כן צריכים לעדכן באופן שוטף את המתכננים השונים בארגון ואת הלקוחות בארגון.

3. ביטחון:
א. ביטחון פיסי הכולל בחירת מבנה מתאים.
ב. נהלים והקצאת סמכויות ותחומי אחריות.
ג. שמירה על קבצים.

4. דיווח:
דיווח לאנשי הארגון על שינויים במערכות, פיתוח מערכות חדשות וכד׳. ״תרגום״ המונחים הטכניים לשפה המובנת לכל אנשי הארגון.

5. ניהול כוח־אדם:
א. הגדרת התפקידים, סמכויותיהם ותחום עיסוקם.
ב. גיוס ובחירת כוח־אדם מתאים.
ג. הדרכת כוח־אדם חדש וקיום קורסים לכוח־אדם קיים.

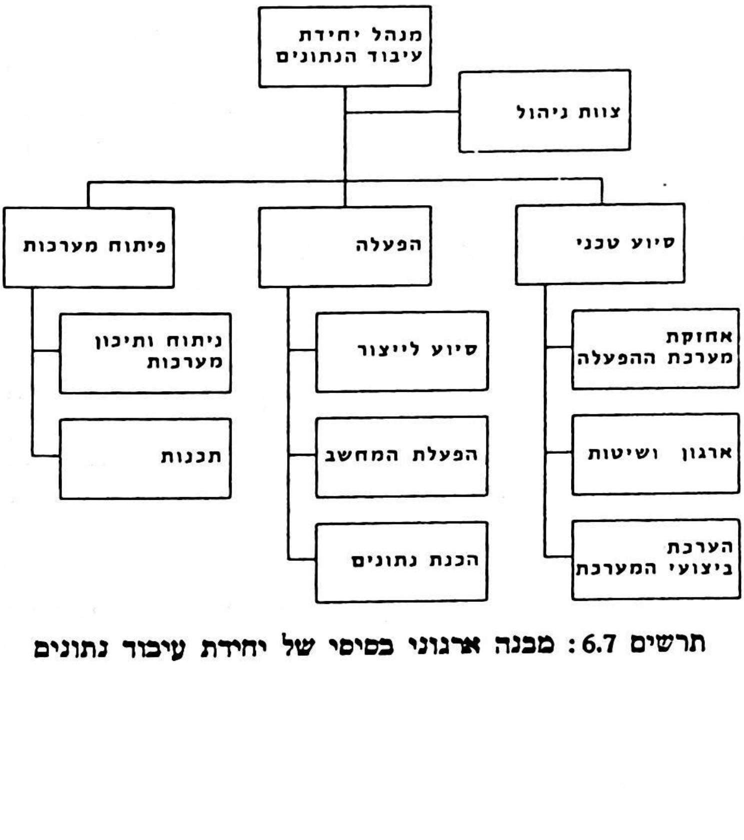
6. ניהול פיננסי:
הקצאת עלות יחידת עיבוד נתונים בין לקוחותיה, תכנון פיננסי, רכישת ציוד וכד׳.

7. מינהל:
נתינת שירותים כלליים הנדרשים בכל פעילות ארגונית:
א. שירותי מזכירות.
ב. רישום ותיוק.
ג. אחזקה וניקיון.

דוגמה למבנה ארגוני בסיסי של יחידת עיבוד נתונים מוצגת בתרשים 6.7. במבנה כזה מחולקת הפעילות לשלושה חלקים תפעוליים ראשיים וקבוצת מינהלה אחת.

מובן, שכל קבוצה מתחלקת בתוכה לפי הצורך.

ביחידות קטנות לעיבוד נתונים, ניתן לאחד את קבוצות פיתוח המערכות והסיוע הטכני לקבוצה אחת.



תרשים 6.7: מבנה ארגוני בסיסי של יחידת עיבוד נתונים

תיאור תפקידים עיקריים במסגרת יחידת עיבוד נתונים

ניתן לחלק את התפקידים במסגרת מערכות מידע למשפחות, אשר כוללות מטלות דומות, אחריות דומה ורקע דרוש דומה. התיאורים המובאים להלן הם כלליים ועשויים להשתנות מארגון לארגון בהתאם לצורך. בארגונים קטנים יתכן שאדם אחד ימלא מספר תפקידים בו זמנית.

ניהול

מנהל עיבוד הנתונים (או מערכות מידע)

מתכנן את כל הפעילויות לעיבוד הנתונים בארגון ומפקח עליהן. מתכנן שיפורים בפעילויות הארגון בעזרת מערכות חדשות או משופרות. אחראי לקיום קשר עם המשתמשים ותומך בשיפור פעילויות הארגון, שיפור השיטות וניצולת טובה יותר של משאבים. מארגן את משאבי עיבוד הנתונים במטרה לתת שירותים יעילים ותועלתיים למשתמשים.

מנהל אדמיניסטרטיבי

מתאם את תקציבי פעילויות עיבוד הנתונים. מיישם מערכת נהלים אשר מכינה הערכות ומדווחת על הוצאות. מפקח על התחייבויות ועל הוצאות בהשוואה לתקציב. מכין את תקציב עיבוד הנתונים עבור מערך הכספים של הארגון. מכין שיטות תמחיר ומנהל את מערכת החיוב עבור השימוש במשאבי מחלקת עיבוד הנתונים. מדווח למנהל מערכות מידע.

מבקר ענ״א (EDP auditor)

תפקידו לבחון את מערכות המידע הפועלות בארגון מהבחינות האלה: אמינות המערכות, שלמות המערכות, אבטחת המערכות, סודיות המערכות, שימוש יעיל במערכות וכד׳. בדרך־כלל אין מבקר ענ״א כפוף למנהל עיבוד הנתונים, אלא להנהלת הארגון.

ניתוח ותיכון מערכות

מנהל ניתוח ותיכון מערכות

אחראי על פרוייקטים לפיתוח מערכות. מתכנן את עבודתם של מנתחי ומתכנני המערכות ומפקח עליה. מתכנן את המשאבים בצורה שתספק את הדרישות והאיכות של המערכות. מבקר את תיעוד המערכות, שלמותן, איכותן והתאמתן לסטנדרטים. מדווח למנהל מערכות מידע.

מנתח מערכות (system analyst)

אוסף ומנתח מידע הקשור בפיתוח ובעדכון מערכות עיבוד נתונים. מתכנן מערכות ושיטות ליישומן ומגדירן. מפקח או מנחה את יישומן של המערכות.

מנתח מערכות פועלות וממליץ על שיפורים. פועל בשיתוף עם המשתמשים לשם איסוף מידע והגדרת מטרות המערכות. מתעד את הממצאים ואת תוצאות הבחינה. מציג את הממצאים, ההמלצות והדרישות בצורה פורמלית. מדווח למנהל ניתוח ותיכון מערכות.

מעצב מערכות (system designer)
מתרגם את הדרישות אשר הוגדרו על־ידי מנתח המערכות לתכנון לוגי, טכני ומעשי. מכין דרישות מפורטות שלפיהן ייכתבו התכניות הדרושות. דרישות אלו כוללות תרשימי זרימה, אשר מציינים קשר בין תת־המערכות, זרימת הנתונים ומשאבים דרושים. מכין תיעוד לנוהלי טיפול בנתוני הקלט ולעיבודים, ומתכנן הפלט לפונקציות השונות. מדווח למנהל ניתוח ותיכון מערכות.

מנתח מדעי הניהול
מיישם שיטות של מדעי הניהול, כמו מודלים מתמטיים, סימולציה, ארגון ושיטות כפתרון לתכנון מערכות מורכבות. מדווח למנהל ניתוח ותיכון מערכות.

ת כ נ ו ת

מנהל תכנות יישומים
מתכנן, מפקח ובוחן את עבודת אנשי התכנות בפיתוח תוכנה חדשה או בעדכון תוכנה קיימת. מתכנן ומיישם שיטות סטנדרטיות לארגון, קידוד ובחינת תכניות. מקיים מעקב אחר תכניות שהושלמו ואחר תכניות הנמצאות בפיתוח, בכל הקשור לשלמות, יעילות, דיוק ועמידה בסטנדרטים. מדווח למנהל מערכות מידע.

מתכנת יישומים (application programmer)
מתכנן ובודק תכניות. כותב תכניות ומכינן להפעלה במחשב. מכין תרשימי זרימה של התכניות הדרושות לעיבוד נתונים. עוזר למעצב המערכות להכין דרישות לקבצים והגדרות לעיבודים. מבצע תכנות לפי סטנדרטים קבועים. בודק ביסודיות את פעולת התכניות המושלמות ואת זיקתן לתכניות אחרות. מדווח למנהל תכנות יישומים.

ה פ ע ל ה

מנהל הפעלה
אחראי לכל הפעילויות הקשורות בביצוע עיבוד הנתונים במחשב. אחראי לפעולת ציוד מערכת המחשב. מכין נהלים הקשורים בביצוע העבודה לשם הבטחת האיכות והביטחון. מנתח שינויים בעומס העבודה ועוקב אחר יישומים חדשים אשר נמצאים בפיתוח, כדי לחזות את תוצאותיהם על התפעול השוטף. מדווח למנהל מערכות מידע.

מנהל משמרת

אחראי להפעלת המערכות במשמרת, כולל למחשב, לתמיכה ולציוד ההיקפי. מקצה מפעילים לפי הצורך ואחראי לשמירה על נוהלי הפעלה, על האחזקה ועל נוהלי גיבוי. מדווח למנהל הפעלה.

מפעיל (operator)

אחראי להפעלת המחשב ולמרכיביו. דואג להרכיב תקליטים, סרטים ונייר במידת הצורך. כמו כן עוקב אחר ההוראות המופיעות על מסך המערכת ומגיב בהתאם. מדווח למנהל משמרת.

תכנות מערכות ותמיכה טכנית

מנהל תכנות מערכות

אחראי לבחירה, ליישום ולהפעלה של מערכת ההפעלה. כמו־כן אחראי לניהול מסדי הנתונים ומערכות התקשורת. עוזר במדידת ובהערכת ביצועים של המערכת ובהכנת תחזיות לגבי עומסים עתידיים. אחראי לפונקציות התמיכה הטכנית לכל היחידות האחרות ביחידת עיבוד הנתונים. מדווח למנהל מערכות מידע.

תכניתן מערכות (systems programmer)

מתמחה בדרך־כלל באחד או יותר ממרכיבי מערכת ההפעלה. אחראי למתן תמיכה ולאחזקה של המערכת הפועלת וכן לאיתור בעיות וניפויין. אחראי גם על הפקת תכניות שירות ספציפיות עבור החלקים האחרים של יחידת מערכות המידע. מדווח למנהל תכנות מערכות.

מינהלן של מסד נתונים (data base administrator)

אחראי לתכנון ולפיקוח של מסדי הנתונים בארגון. כמו־כן אחראי על הגדרת מסדי נתונים, סגמנטים ושדות; בחירת שיטות גישה פרמטרים וסוגי קשרים. אחראי לפיקוח על סודיות הנתונים ולשלמותם, כולל לתיעוד הקשור במסדי הנתונים. מדווח למנהל תכנות מערכות.

מנהל תקשורת

אחראי לתכנון מערכת התקשורת וליצירתה; אחראי לבחירת הציוד, כגון מסופים, לתכנון השדרים ולמבניהם ולכיוונון מערכת התקשורת. מדווח למנהל תכנות מערכות.

מנהל ספרייה טכנית

אחראי לניהול ולארגון הספרייה הטכנית, הכוללת ספרות טכנית ותיעוד המערכות. כמו־כן אחראי על קביעת תקנים לתיעוד ולפיקוח. מדווח למנהל תכנות מערכות.

הכנת תיאור תפקידים

תיאור התפקיד מהבחינה הפורמלית מסייע להגדרת התפקידים והאחריות של כל תפקיד בסביבת מערכות המידע. היתרונות ביישום תיאור התפקיד הן:

- מגדיר את מיקומו של התפקיד בתוך הארגון שאליו הוא שייך
- משמש כבסיס להשוואת החשיבות היחסית, הרקע והניסיון הנדרש מכל תפקיד, ומשמש לקביעת דרגת השכר המתאימה
- ניתן לשימוש ככלי הדרכה: העובדים ידעו מראש מה עליהם לעשות, למי הם מדווחים וכיצד יוערכו
- מגדיר מסלולי קידום
- משמש כבסיס לתכנון ולתקצוב

המבנה הבסיסי של תיאור תפקיד צריך לכלול לפחות את הסעיפים האלה:

- שם התפקיד
- רמת השכר (דרגה או תחום הדרגות)
- המנהל הישיר (למי מדווח תפקיד זה)
- שמות התפקידים המנוהלים ע״י תפקיד זה
- מספר האנשים המנוהלים בצורה ישירה על־ידי תפקיד זה
- רשימת האחריויות והסמכויות
- קשר עם גורמים מחוץ למערכות המידע
- ניסיון ורקע דרוש
- מדדים להערכה.

סוגים שונים של מערכות מידע והתאמתם לארגון

פעילויות רבות מתרחשות כל יום בחיי הארגון. כל פעילות יוצרת תנועה אחת או יותר. כל תנועה משתקפת ברשומה (מסמך מקור). התנועות מהוות נתונים גולמיים שניתנים לניתוח בדרכים שונות, כך שמסופקות דרישות המידע הייחודיות של רמות תפעול וניהול שונות בארגון.

מערכות לעיבוד תנועות — מע״ת (transaction processing systems) — נושאת את החלק העיקרי בעומס של מערכת המידע. הן רושמות ומאחסנות את רוב הנתונים הדרושים למערכת המידע ומייצרות את רוב הדוחות והמסמכים

הנדרשים ע״י המשתמשים התפעוליים. בכל ארגון קיימות מספר מערכות כאלו, כל אחת משרתת יישום ייחודי (מלאי, כספים וכד׳). חלק מהמערכות קשורות ביניהן, אם באמצעות שיתוף בקבצי־אב ואם עקב העובדה שפלט של מע״ת אחת משמש כקלט למע״ת אחרת.

מערכות החלטה מובנית — מה״מ (structured decision systems) — הן מערכות מידע המקבלות החלטות מובנות (מתוכנתות). יש להן נגישות (access) למסדי הנתונים המאוחזקים על־ידי המע״ת והן כוללות מודל החלטה אחד או יותר המקודד בתכנית מחשב. מודל החלטה כזה מייצג תהליך קבלת החלטה אנושי שהוא מתוכנת לחלוטין.

בשונה ממע״ת וממה״מ מערכת תומכת החלטה — מת״ה (decision support system) — תומכת בפתרון בעיות בלתי־מובנות הטיפוסיות לעולמו של המנהל הבכיר בארגון. מת״ה יכולה לתמוך במנהל בכל אחד משלבי קבלת ההחלטה שלו ומגבירה את הישגיות המנהל המשתמש בה. בשונה ממה״מ, אין מת״ה מקבלת החלטות בעצמה.

מערכות לעיבוד תנועות

פעילויות רבות בארגון מתרחשות תקופתית: שכר משולם בסוף החודש, ציונים ניתנים בסוף השליש. המסמכים הרושמים פעילויות אלו נאספים באצווה (batch), האצווה מעובדת וקבצי־אב (קובץ עובדים, קובץ סטודנטים) מעודכנים בתאריך קבוע. אצווה היא בדרך־כלל הדרך הזולה ביותר לביצוע עיבוד תקופתי, כאשר הקבצים אינם צריכים לשקף את כל התנועות עד לרגע האחרון.

מחזור עיבוד התנועות, המבוצע ע״י כל מע״ת, מורכב מחמישה שלבים: (1) רישום ואיסוף נתונים, (2) הסבה ובדיקת נכונות, (3) אימות הנתונים, (4) עדכון ועיבוד קבצים, (5) דיווח.

רישום ואיסוף נתונים

ביצוע תנועה (מכירה, קניה) נרשם בדרך־כלל על גבי מסמך או טופס. קליטת הנתונים היא צעד הכרחי ראשון בעיבוד תנועה. לנוהלי הרישום השפעה מכרעת על מהירות הרישום ודיוקו ועל תהליך איתור השגיאות ותיקונן אחר־כך.

איסוף הנתונים הוא לעתים קרובות אחד המרכיבים היקרים ביותר בתפעול מערכת מידע, ולכן בחירת היתקני האיסוף (היתקני קלט) הנה אחת ההחלטות החשובות ביותר בעת תיכון מערכת מידע. קיים מגוון רחב של היתקני איסוף, השונים זה מזה באמצעי רישום וחישת נתונים, רמת המיכון, מהירות, ניידות, עלות ודיוק. הטופס עליו נרשמת תנועה יכול להימלא ידנית, להיתקתק, להידפס מראש בחלקו (preprinted), או להיקדד. טפסים מודפסים מראש וקידוד מקטינים את

ההסתברות לרישום שגוי. במערכות מודרניות קיימת האפשרות שה"טופס" יהיה מוצג על גבי מסך, והנתונים המוקשים ל"טופס" יקלטו מיידית במחשב.

הסבה ובדיקת נכונות (data conversion and verification)

הסבת נתונים היא תהליך של המרת נתונים מאמצעי קריא לאדם (למשל, טופס בכתב־יד) לאמצעי קריא למחשב (למשל, סרט מגנטי). תנועות כתובות עוברות תהליך של קליטת נתונים שמסב את הנתונים לקלט קריא למחשב. הרשומות המוסבות נהפכות לקובץ תנועות.

מע"ת טיפוסית יכולה להשתמש בהיתקנים שונים כדי לעמוד במגוון של דרישות ומקורות נתונים: מכונות ניקוב, היתקני הקשה (הקשה לסרט, לתקליט, לתקליטון וכד'), מסופים מקוונים. הפלט הקריא למחשב של היתקנים אלו יכול לכלול כרטיסים מנוקבים, סרט מגנטי (או תקליט, תקליטון וכד'), מסמכים קריאים לקורא אופטי וכד'. קיים כיום שימוש נרחב במסופים לצורך קליטה ישירה של תנועות למערכת המחשב. במקרים אלו לא נרשמים מסמכי מקור ידניים והשלבים של רישום הנתונים והסבת הנתונים הנם משולבים: הרישום המקורי קריא למחשב. דוגמאות לכך הן בנקומטים ומסופי נקודות מכירה (POS terminals) בסופרמרקטים.

פעילויות הסבת הנתונים עלולות ליצור שגיאות. התהליך המוודא שהנתונים שנרשמו על מסמך מקור הוסבו באופן נכון למסמך הקריא למחשב הוא תהליך "בדיקת נכונות". בדיקה זו ניתנת לביצוע ע"י הקשה חוזרת של מסמך המקור או ע"י בדיקה בעין; עם היתקני ההקשה והמסופים הקיימים כיום, ניתן לבצע בדיקת נכונות באמצעות תכנית המאוחסנת ביחידת הזיכרון של ההיתקן, אשר מאתרת שגויים במהירות ודורשת מיידית הקשה חוזרת של מסמך המקור או חלקים ממנו.

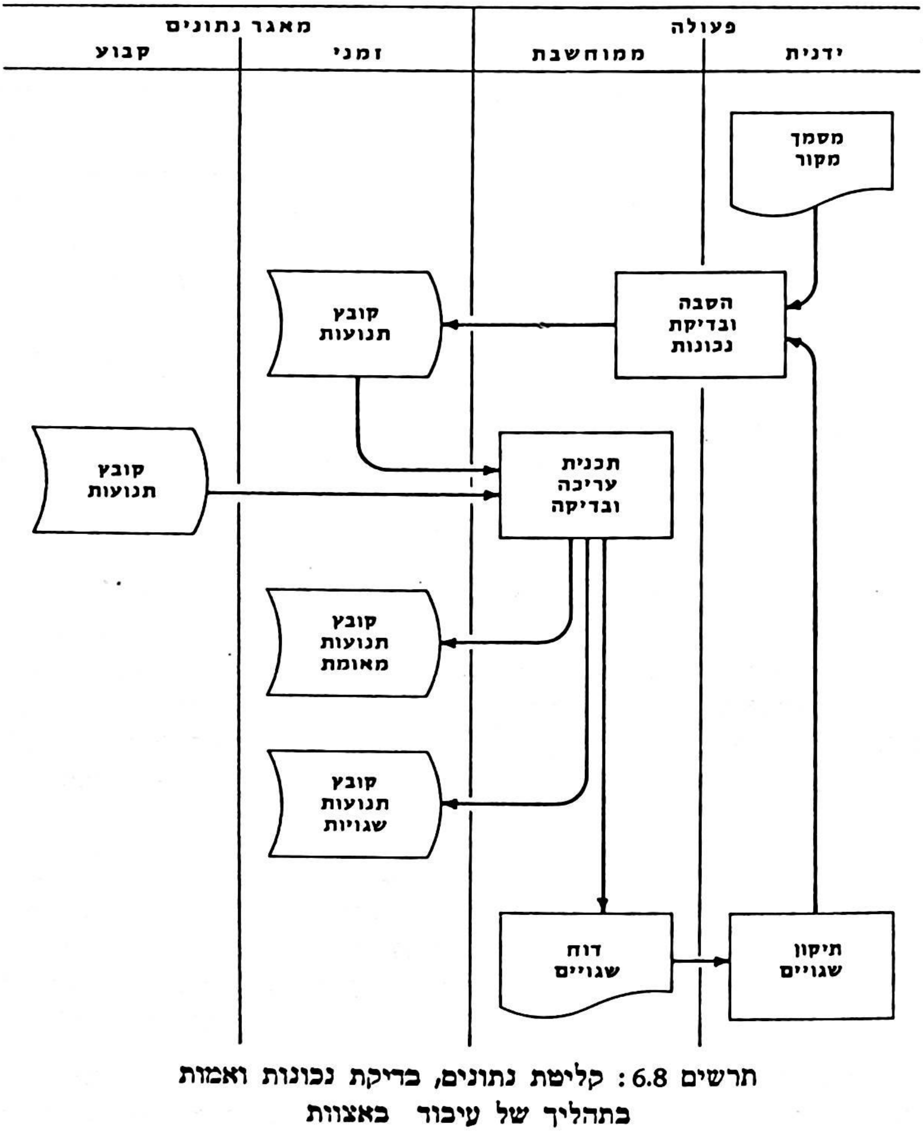
אימות נתונים

בדיקת נכונות עוסקת בדיוק ובשלמות של מסמכי מקור שהוסבו; היא יכולה להתבצע ידנית או באמצעות תכנית מחשב ואינה דורשת השוואה לרשומות בקובץ אב. לעומת זאת, אמות הוא תמיד תהליך ממוחשב הכרוך בקובץ תנועות שרשומותיו כבר נבדקו, והוא בודק יחסים בין רשומות תנועות ורשומות קובץ אב.

אמות נתונים עוסק בבדיקת רשומות תנועה בהשוואה לקבוצה של קריטריוני אמות שנועדו לאתר סוגים מסוימים של אי־דיוק, אי־שלמות או חוסר הגיון. בדיקות אמות מבוצעות בדרך־כלל על תווים ושדות בודדים או על הרכב של

כמה שדות מרשומת התנועה, בפני עצמם, או ביחד עם התוכן של רשומות מאחד או יותר קבצי־אב.

תרשים 6.8 מתאר את תהליך האמות המבוסס על עיבוד באצוות.



תרשים 6.8: קליטת נתונים, בדיקת נכונות ואמות בתהליך של עיבוד באצוות

תכנית מחשב (הנקראת בדרך־כלל תכנית עריכה), קוראת את רשומות התנועה ובוחנת שגויים אפשריים. התכנית יכולה למשל לבדוק אם מספר הלקוח ברשומת תנועה זהה למספר לקוח פעיל הנמצא בקובץ אב לקוחות. התכנית מפיקה קובץ תנועות חדש המכיל רשומות מאומתות בלבד וכן רשימת רשומות המכילות שגויים.

הרשומות שנדחו מתוקנות, נקלטות ונבדקות שנית. התהליך מסתיים כאשר תנועות הקלט נקיות משגיאות. הרשומות המכילות שגיאות נרשמות על קובץ ביניים למעקב — כדי לוודא שיתוקנו או שיבוטלו סופית. ע״י כך מוודאת תכנית העריכה שהרשומות השגויות יתוקנו ולא יאבדו, ושעיבודן לא יתעכב. טכנולוגיות מידע מודרניות מאפשרות שילוב שלבי הרישום, בדיקת הנכונות והאמות באמצעות רישום ישיר של נתונים לתוך מחשב זעיר או באמצעות היתקן קליטת נתונים הניתן לתכנות ("חכם"). כללית, ככל שהיתקן קליטת הנתונים מתוחכם, כך גבוה יותר השילוב בין השלבים. ההפרדה בין השלבים נקבעת על־ידי החלופות הטכנולוגיות והכלכליות העומדות בפני כל ארגון.

עדכון ועיבוד קבצים

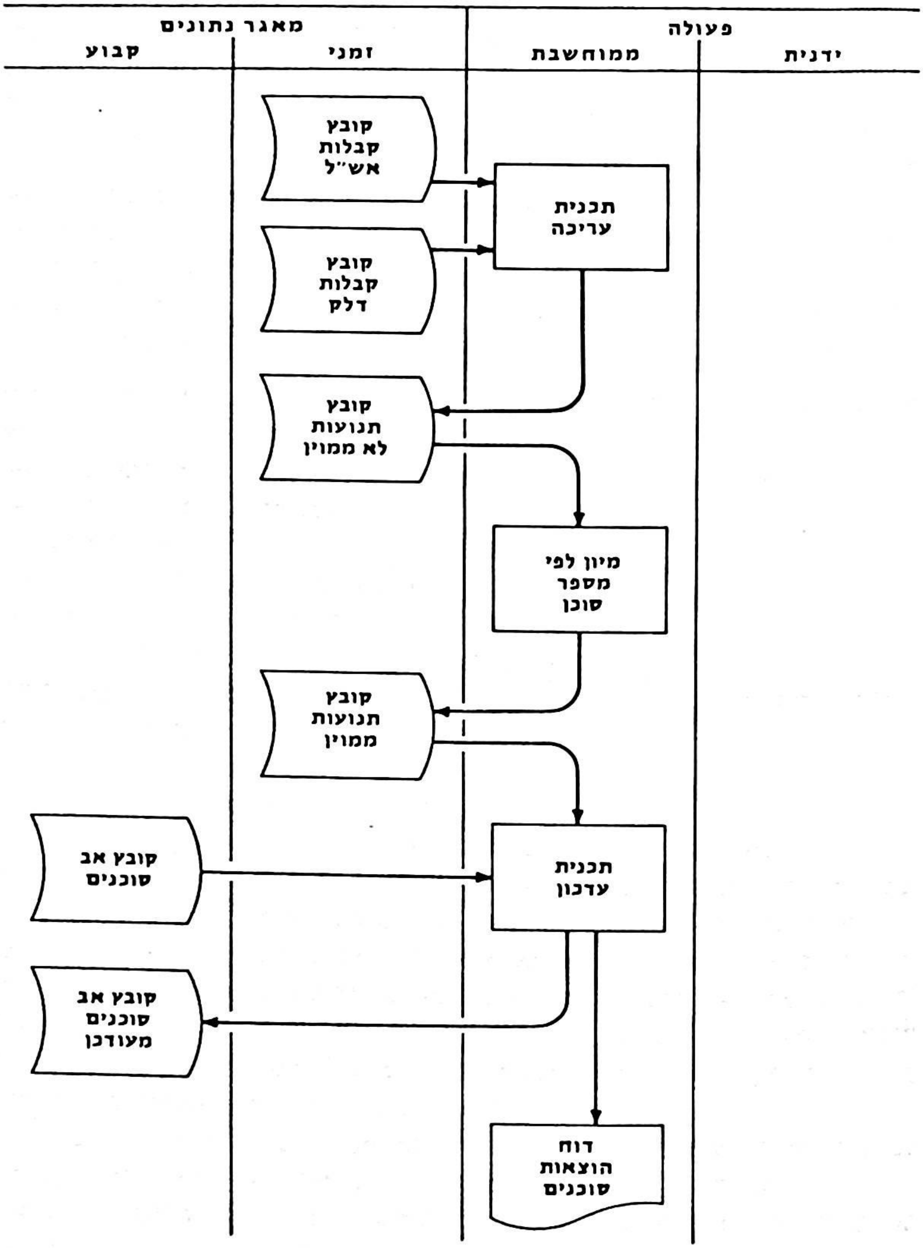
עיבוד הקבצים מתבצע כאשר קובץ תנועות הקלט אומת, תוקן ולא נמצאו בו יותר שגיאות. עיבוד הקובץ יכול להתבצע בכמה אופנים ויכול להיות מעורב בקובץ זה בלבד, או בקובץ אב אחד או יותר.

מיון (sorting) — דוגמה שכיחה לעיבוד הקובץ בלבד. מיון פירושו סידור הרשומות בקובץ בסדר מוכתב מראש לפי שדה מסוים ברשומה (למשל, לפי מספר לקוח). בעיבוד קובץ התנועות יכול להיות מעורב גם קובץ אב שממנו נשלפים נתונים שונים. דוגמה לכך היא עיבוד תנועות מכירה לצורכי חיוב לקוחות: רשומות התנועה יכולות להכיל מספרי לקוח בלבד; שם הלקוח וכתובתו נשלפים מקובץ אב לקוחות.

עדכון (updating) — דוגמה נוספת לעיבוד קובץ תנועות עם קובץ אב אחד או יותר. קובץ אב חייב להתעדכן ע״י שינוי שדות ברשומות, הוספת רשומות חדשות וביטול רשומות ישנות. למשל, בעדכון קובץ־אב סטודנטים יש לקלוט או לשנות ציונים, להוסיף סטודנטים חדשים ולבטל רשומות של סטודנטים שהשלימו לימודיהם.

שיטת עדכון קבצי־אב מותנה בהיתקני האחסנה החיצוניים (סדרתיים או ישירים), בצורת ארגון הקובץ (סדרתי, טבלתי או ישיר) ובשיטת הגישה לרשומות בקובץ (סדרתית, סדרתית באמצעות אינדקסים או ישירה). אלו תוארו בפרק 4.

תרשים 6.9 מציג עדכון סדרתי של קובץ סוכני מכירות ע"י קובץ תנועות אש"ל ודלק של הסוכנים.



תרשים 6.9 : תהליך רישום הוצאות ועדכון קובץ־אב סוכנים

ד י ו ו ח (reporting)

שלב הדיווח מהווה את הממשק (interface) בין המע"ת לבין המשתמשים. לא כל הפלטים של המע"ת מדווחים (או מוצגים). חלק גדול מהפלטים יכול להיות מאוחסן במסד הנתונים (קבצי־אב שעודכנו). מע"ת משכורת, למשל, יכולה לייצר מספר פלטים, ביניהם קבצים (קובץ אב משכורות מעודכן, קובץ המחאות), דיווחים מודפסים (דיווחי מס) ומסמכים (המחאות).

מע"ת מייצרות שלושה סוגי פלטים עיקריים:

— דיווחים מתוכננים מראש מבחינת תבנית ותוכן, למשל מאזנים
— תשובות מתוכננות מראש לשאילתות (למשל דוח חריגים על לקוחות שחרגו ממסגרת האשראי)
— מסמכי תנועה (או מסמכי פעולה), למשל המחאות, חשבונות חשמל, הזמנות מספקים.

קיימות שיטות דיווח שונות ומגוון של היתקני דיווח, השונים מבחינת עלויות ותועלות:

— טפסי נייר מודפסים מראש
— פלט טפסי נייר מהמחשב
— כרטיסים מנוקבים
— פלט "קשה" (מודפס) של מסופים
— פלט מיקרופילם של מחשב
— פלט "רך" (מוצג על מסך) של מסופים.

הפלטים ניתנים להצגה במספר צורות: מילולית, טבלתית (לוחות) וגרפית (תרשימים, עקומות, שרטוטים).

מ ע ר כ ת מ ק ו ו נ ת ל ע י ב ו ד ת נ ו ע ו ת (on-line tps)

עיבוד תנועות מקוון יכול לספק אמות נתונים מיידי, או עדכון ועיבוד מיידי, או דיווח מיידי, או שילוב של כל השלבים הללו. בסביבה מקוונת יש למשתמש נגישות ישירה לקבצים ולתכניות מחשב באמצעות מסופים וקווי תקשורת.

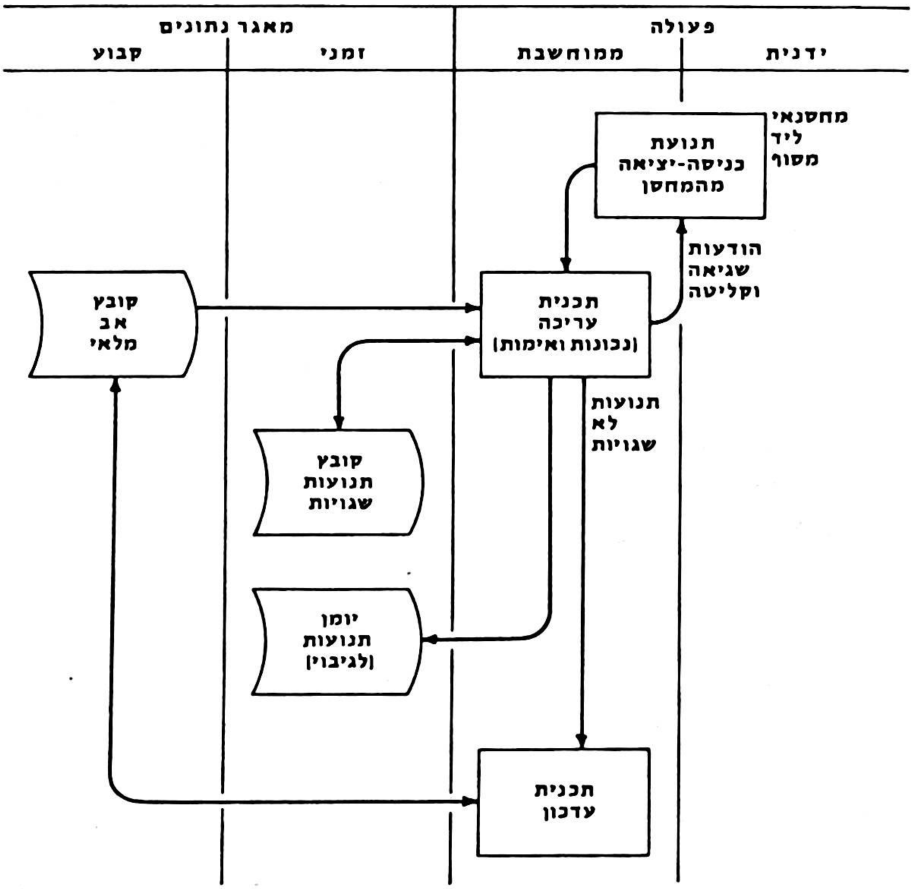
במערכת מסוג זה רצף הגעת התנועות הוא אקראי וכל מסוף הוא מקור קלט נפרד. התנועות מעובדות בצורה שוטפת מיד עם קליטתן — ולא תקופתית. הקבצים מקוונים ובעלי גישה ישירה לרשומות, וזמינים כל הזמן לצורכי אמות ועדכון. תנועות שגויות מתגלות מיד עם הקשתן, ניתנות לתיקון ע"י האדם העוסק בהזנת הנתונים ואינן מעוכבות לעיבוד תקופתי.

קבצי־האב מעודכנים באופן שוטף ומשקפים, כמעט מיידית, את התרחשות כל האירועים בחיי הארגון (בתנאי שהתנועות מוזנות מיידית למערכת).

יתרונות אלה של עיבוד תנועות מקוון דורשים עלויות ניכרות. זמן מחשב מיוחד דרוש לעיבוד ולשימוש המסופים; התוכנה הבסיסית והתוכנה היישומית מורכבות יותר מאלו הכרוכות בעיבוד באצוות; דרושה השקעה רצינית ברכישת מסופים

ובתפעולם, בקווי ובהיתקני תקשורת, בהיתקני אחסנה של גישה ישירה, במחשבים גדולים ובאמצעי בקרה.

לעתים מקיף העיבוד המקוון רק חלק ממחזור עיבוד התנועות, בעיקר את שלבי האיסוף, ההסבה ובדיקת הנכונות: התנועות מוזנות ישירות באמצעות מסופים, נבדקות מיידית ומקובצות באצוות לעיבוד במועד מאוחר יותר. היתרונות של שיטה זו לעומת עיבוד באצוות מסורתי הם הקליטה המיידית של תנועות בצורה קריאה למחשב והמשוב המיידי למשתמש לצורכי תיקון שגויים. תרשים 6.10 מציג עיבוד מקוון של תנועות מלאי.



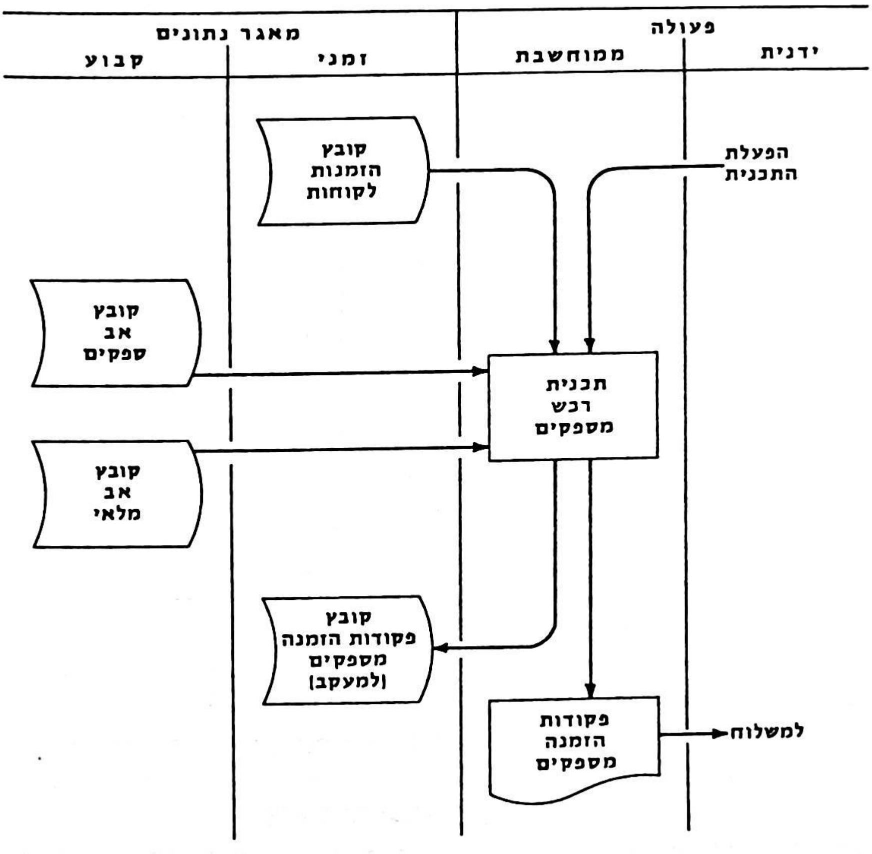
תרשים 6.10: עיבוד תנועות מקוון של כניסות ויציאות מלאי

מערכות החלטה מובנית

מה״מ היא מערכת מידע שמקבלת החלטות מובנות (מתוכנתות). בעוד שמע״ת תומכת בפעילויות הדרג התפעולי בארגון, מקבלת מה״מ החלטות בתחום

המנהלים התפעוליים והטקטיים. מה"מ ניתנות לפיתוח רק אחרי שקיימות כבר מע"ת, כיון שההחלטות האוטומטיות המתקבלות ע"י מה"מ הן תוצאה של עיבוד נתונים המעובדים ע"י מע"ת. לדוגמה, מה"מ שקובעת כמויות הזמנה מספקים אינה יכולה לפעול ללא מערכת לעיבוד תנועות מלאי, הקולטת תנועות מלאי ומעדכנת קובץ אב מלאי. מה"מ זו יכולה לפעול כיישום בלתי תלוי המופעל ע"י התערבות אנושית (מנהל המחסן), או כחלק משלבי העיבוד במחזור עיבוד תנועות־המלאי.

כל מה"מ כוללת תכנית מחשב אחת לפחות שמורה למחשב לעבד נתונים רלוונטיים באמצעות מודל החלטה אחד או יותר. נניח, לדוגמה, מה"מ שמייצרת אוטומטית פקודות רכש לספקים. המערכת נשענת על תכנית מחשב שמחשבת רכש נדרש של פריטי מלאי כפונקציה של הזמנות לקוחות, מלאי ביטחון של פריט, יתרת המלאי של הפריט וכמות הפריט הנמצאת בהזמנה מהספק. תרשים 6.11 מציג מערכת זו.



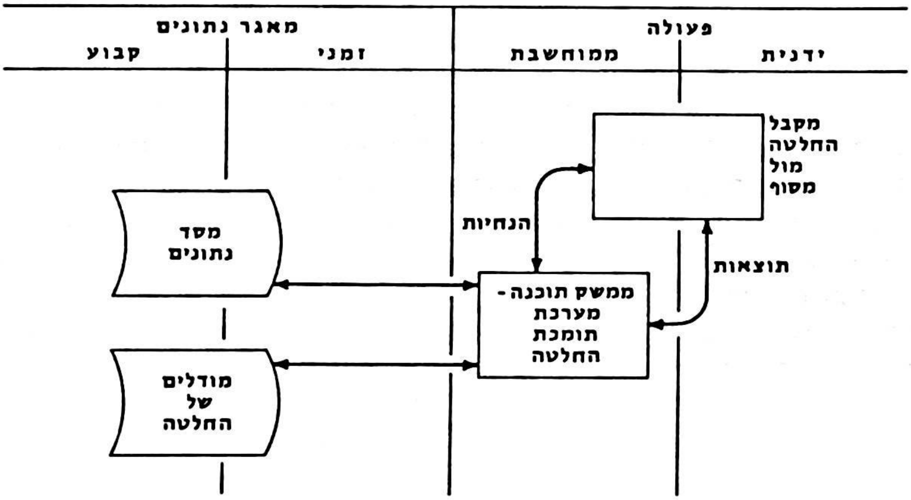
תרשים 6.11 : מה"מ להזמנת פריטי מלאי מספקים

המה"מ המתוארת בתרשים מקבלת החלטות רכש שנתקבלו בעבר ע"י הקניין של הארגון, והיא עושה זאת מהר ובדייקנות. היא משתמשת בנתונים שנקלטו ועובדו ע"י המע"ת של הזמנות לקוחות, המע"ת של המלאי והמע"ת של הספקים.

מערכת תומכת החלטה

מת"ה עוזרת למנהל (בדרך־כלל הבכיר) בפתרון בעיות בלתי מובנות. היא נשענת על שיקולי מקבל ההחלטה האנושי ועל שפיטתו בכל שלבי פתרון הבעיה — הגדרת הבעיה, בחירת נתונים רלוונטיים לבעיה, בחירת פתרונות חילופיים והערכת החלופות.

למת"ה שלושה רכיבים חיוניים: (1) נתונים (2) מודלים של החלטה ו־(3) מקבל החלטה. אלו מוצגים בתרשים 6.12.



תרשים 6.12: רכיבים חיותיים של מערכת תומכת החלטה

התוכנה הבסיסית הנדרשת למת"ה ניתנת להשגה מספק חיצוני, או נשענת על פיתוח עצמי של צוות מקצועי או של מקבלי ההחלטות. התוכנה דרושה כדי לשוחח עם מסד הנתונים ומסד המודלים (תכניות מחשב). ניהול מסד הנתונים נעשה ע"י מערכת מסדי נתונים. תכניות המחשב היישומיות הנמצאות במסד המודלים יכולות להיות חבילות קנויות או פרי פיתוח עצמי.

מת"ה נדרשת לאמצעי חומרה ספציפיים. החלטות המתקבלות בעזרת מת"ה דורשות נתונים זמינים. לעתים יש להריץ מודל החלטה מספר פעמים, עם שינוי מסוים בפרמטר(ים) של המודל לצורך בדיקת רגישות, לפני שהמשתמש מוכן לקבל החלטה. פירוש הדבר תגובה מהירה של המערכת באמצעות עיבוד מקוון.

התועלת העיקרית ממת״ה נובעת, אם כך, מיכולת מקבל ההחלטות לקבל משוב מידי באמצעות גישה ישירה למסד הנתונים ומסד המודלים. לדוגמה, במערכת כזו יהיו צגים שיכוונו את המנהל בעזרת תפריט של קבצים ותכניות העומדים לרשותו; המנהל יצטרך רק לבחור את הקובץ והתכנית הדרושים כדי להתחיל בשרשרת האירועים שיובילו להחלטה הסופית שלו.

מבנים אפשריים של מערכות מידע ממוחשבות בארגון

מערכות מידע ממוחשבות מתחילות מסורתית ביישומים של עיבוד תנועות, יכולות לעבור למערכות החלטה מובנית, ועתה לפתח אולי חוליות תומכות החלטה. לאורך נתיב זה של שירותים ההולכים ומתוחכמים, יכולות פעילויות רבות של מערכת המידע לשנות צורה. שינויים אלו נוגעים לחומרה, לתוכנה, ליישומים, לתכנות, לכוח־אדם, לניתוח־מערכות, לבקרה, לתפעול, למסדי נתונים וכדומה. הפעילויות הללו יכולות להיות מרוכזות, מבוזרות או מפולגות.

מערכת מרוכזת (centralized system) — מערכת מרוכזת לחלוטין מבצעת את כל העיבודים באתר מחשב יחיד; מאחזקת מסד נתונים מרכזי אחד; נוקטת בפיתוח מרוכז של יישומים ואספקה מרכזית של שירותים טכניים; קובעת קדימויות פיתוח במרכז ומקצה משאבי מחשב מהמרכז. המשתמשים של המערכת יכולים לקבל שירותי מיחשוב ע״י העברה פיסית או אלקטרונית של נתוני קלט ופלט.

מערכת מבוזרת (decentralized system) — במערכת מבוזרת לחלוטין אין בקרה מרכזית של פיתוח יישומים; אין קשרי תקשורת בין יחידות חישוב אוטונומיות; קיימים מעבדים ומסדי נתונים עצמאיים באתרים שונים. כל יחידה בארגון מממנת את פעילויות עיבוד המידע שלה ואחראית לחלוטין לפיתוח מערכת המידע שלה ולתפעולה.

יתרונות של מערכות מרוכזות

היתרונות העיקריים לריכוז מערכות מידע:

— יתרונות לגודל אפשריים עם מערכות ריכוזיות גדולות, המקטינות את הצורך בריבוי חומרה, תוכנה, אתרים, כוח־אדם ומסדי נתונים
— אפשרויות טובות לגייס כוח־אדם מיומן וביצוע תכניות הדרכה
— ניצול טוב ביותר של משאבי המערכת ואיזון עומסי עיבודים
— נוחות הבקרה של מערכת המידע ע״י הנהלת הארגון
— אמצעים טובים לבטיחות, לגיבוי ולשחזור
— אכיפת תקנון באמצעות פיתוח אחיד של יישומים
— ניהול טוב של פעילויות מערכת המידע, כיוון שכל המשאבים ניתנים לתיאום מהמרכז

— יכולת לספק ליחידות ארגוניות קטנות או רחוקות נגישות למשאבים ולישומים שאחרת לא יהיו נגישים עבורן
— התאמה טובה של מערכת המידע לתכנון הכלל-ארגוני.

יתרונות של מערכות מבוזרות

היתרונות העיקריים לביזור מערכות:

— הנעה ומעורבות גבוהים של המשתמשים
— בקרה טובה של המשתמשים
— אפשרות להתמחות של משתמשים או פונקציות ספציפיים
— פחות בעיות של תיאום בין תחומים ארגוניים
— עלויות ונפחים מוקטנים של העברת נתונים פיסית או בתקשורת
— זמני תגובה מהירים למשתמשים
— אמינות גבוהה של מערכות מקומיות כתוצאה ממורכבות נמוכה
— התאמת המערכות המקומיות לצורכי המשתמשים
— אחריות תקציבית ותמחירית ישירה של מערכות מקומיות
— הקטנת הסיכון לנזק מוחלט בשירותי עיבוד מידע עקב ריבוי אתרים
— תכנות וטכנולוגיות מחשב פשוטים יחסית
— נוחיות להתקנת שינויים במערכת מקומית ללא השפעה על מערכות מקומיות אחרות.

מערכת מפולגת (distributed system)

הטיעונים המצדיקים מערכת מידע מרוכזת יכולים לשמש כטיעוני נגד למערכת מבוזרת ולהיפך. המצדדים במערכת מרוכזת מתמקדים ביעילות המושגת כתוצאה מיתרונות לגודל. המצדדים במערכת מבוזרת מעדיפים את ההישגיות הנובעת ממערכות אוטונומיות המותאמות לצורכי המידע של משתמשים מקומיים.

טכנולוגיות המידע הנוכחיות (מחשבים זעירים, תקשורת, תוכנה בסיסית ויישומית מתוחכמות, מסדי נתונים, עצמת עיבוד זולה ואמצעי אחסנה זולים) מאפשרות מבנים של מערכות מידע שאינם בהכרח מרוכזים או מבוזרים לחלוטין. במבנים אלו ניתן לממש יתרונות רבים של שני המבנים הקיצוניים הללו ולהימנע מחסרונות רבים. מערכות אלו מכונות מערכות מידע מפולגות. כולן נמצאות בתחום שבין ריכוז מוחלט לביזור מוחלט.

ההגדרה השכיחה של מערכת מפולגת מתייחסת לפיזור החומרה והנתונים בין אתרים שונים בארגון, הקשורים ביניהם במערכת תקשורת. הגדרה זו מתעלמת ממגוון רחב של פעילויות אחרות הכרוכות בניהול עיבוד המידע. פעילויות אלו מסווגות לפעילויות ביצוע ופעילויות בקרה.

פעילויות ביצוע עוסקות בפיתוח (ניהול נתונים, תיכון מערכות, הדרכה, פיתוח יישומים) ובתפעול (אחזקת יישומים, עיבוד, קליטת נתונים). פעילויות בקרה עוסקות בתכנון ובבקרה של משאבי מערכת המידע (קביעת קדימויות, בטיחות, תקצוב, תזמון מטלות). כל פעילות יכולה להיות מבוצעת ע״י צוות מרכזי או

ע״י משתמש. פילוג הפעילות קשור ברמת האחריות המואצלת למשתמש על הפעילות. במערכת מידע מפולגת מאוצלת האחריות ביחס לפעילות אחת לפחות במידה מסוימת למשתמש. מכך נובע שקיימים מבנים פוטנציאליים רבים של מערכות מידע מפולגות. להלן כמה מבנים חילופיים שכיחים:

1. עיבוד מפולג בעל מסד נתונים מרכזי: חלק מהעיבוד מתבצע במחשבים קטנים מקומיים (למשל, קליטת ואמות נתונים), אך ללא תמיכה של מסד נתונים מקומי.
2. מערכת הירככית בעלת מסדי נתונים מקומיים: העיבוד הוא מקומי ומסד נתונים מקומי מעודכן ע״י התנועות המקומיות, ותקופתית גם ע״י מסד הנתונים המרכזי. נתונים מתומצתים נשלחים למחשב המרכזי לעיבוד גלובלי ולעדכון מסד הנתונים המרכזי.
3. מסד נתונים מפולג בעל שיתוף מוגבל: בנוסף לעיבוד מקומי ולמסד נתונים מקומי, יש ליחידה מקומית נגישות למסדי נתונים מקומיים אחרים במקרים חריגים (למשל, לבדיקת יתרת עו״ש של לקוח בנק הרוצה למשוך כסף מסניף שאיננו שלו).
4. מערכת הירככית מפולגת רבת־רמות: כל מעבד מקומי במערכת המפולגת מהווה צומת מרכזי של מערכות מפולגות נמוכות ממנו.
5. מערכת מפולגת לחלוטין: מורכבת ממעבדים אוטומטיים רבים וממסדי נתונים מפולגים בעלי סטאטוס בקרה שווה. זהו מבנה בלתי הירככי חסר בקרה מרכזית וחסר מחשב יחיד בעל תפקיד מרכזי. המערכת מאפשרת שיתוף משאבים (בעיקר של תכניות ושל נתונים) בין אתרי המערכת.

חלק רביעי

ענף המחשבים ומגמות לעתיד

פרק 7.

ענף המחשבים ומגמות לעתיד

ענף המחשבים כולל חברות רבות העוסקות בייצור ובשיווק ציוד, מוצרי תוכנה ושירותים הקשורים בפיתוח וביישום מערכות מידע ממוחשבות.

הפעילות המסחרית בענף מתחלקת לחמישה תחומים עיקריים. הגבולות בין תחומי הפעילות אינם חד־משמעיים; רוב החברות פעילות במספר תחומים.

תחומי הפעילות העיקריים בענף:

— יצרני/משווקי חומרה (מחשבים וציוד היקפי)
— יצרני/משווקי תוכנה
— יצרני/משווקי מערכות ממוחשבות (חומרה ותוכנה)
— לשכות שירות
— חברות ייעוץ.

יצרני/משווקי חומרה (hardware manufacturers)

חברות אלה מייצרות ומשווקות מחשבים וציוד היקפי (מסופים, מדפסות, אמצעי אחסון מגנטיים וכד׳). בדרך־כלל הציוד שמוכרות חברות אלה כולל את התוכנה הבסיסית (מערכת הפעלה, בקרים, וכד׳), אך איננו כולל את התוכנה היישומית. יצרני המחשבים מאופיינים על ידי גודל מערכות המחשב שהם משווקים. קביעת גודל מערכת המחשב איננה חד־משמעית. קיימות מספר תכונות המקובלות בענף, המשמשות לאפיון מערכת המחשב. טבלה 7.1 מתארת את התכונות למשפחות המחשבים. חלק מהיצרנים מתמחה בייצור מחשבים ממשפחה אחת; יצרנים אחרים מייצרים מספר משפחות.

טבלה 7.1: תכונות של משפחות מחשבים

משפחות מחשבים	גדולים		ביניים		קטנים
התכונה	ענק	גדול	בינוני	מיני	מיקרו
מחיר מערכת המחשב ב־$	1,000,000.— 12,000,000	200,000— 1,000,000	50,000— 250,000	10,000— 100,000	500— 20,000
קבולת זיכרון ראשי (בתים)	גדול מאד 2־16MB	גדול 1־8MB	בינוני 256KB־2MB	בינוני 64־512KB	קטן 4־256KB
יכולת לתמוך בציוד קלט/פלט (מסופים, מדפסות וכו׳)	רבה	רבה	טובה	סבירה	מוגבלת
מהירות עיבוד	מהירה מאוד	מהירה	בינונית	בינונית	אטית
שפות תכנות ותכנה בסיסית	מגוון רחב	מגוון רחב	מגוון	מוגבל	מוגבל
תנאי סביבה (מזוג, לחות, חשמל וכד׳)	מבוקרת מאוד	מבוקרת מאוד	מבוקרת יחסית	אין דרישות מיוחדות	אין דרישות מיוחדות
עצמת חישוב	גבוהה מאוד	גבוהה מאוד	גבוהה	בינונית	נמוכה

יצרני מחשבים גדולים (mainframe computers)
בתחום זה קיים מספר קטן של יצרנים גדולים. כניסת חברות חדשות לתחום זה מוגבלת, עקב השקעות הון עצומות. יתרונות להיקף הגדול המוקדש למחקר ולפיתוח חומרה ותוכנה.

יצרני מחשבים ביניים (mini computers)
קיים מספר גדול של חברות גדולות, בינוניות וקטנות. בתחום זה חל גידול רב בשנים האחרונות, אך מסתמנת נטייה של התייצבות.

יצרני מחשבים קטנים (micro computers)

מספר החברות גדול מאוד. תחום זה מצוי בגידול מדהים הן מבחינת ההיקף הכספי והן מבחינת מספרי החברות. כניסת חברות חדשות קלה, משום מחסום הון נמוך ושוק פוטנציאלי עצום. לעומת זאת התחרות רבה והשינויים הטכנולוגיים תכופים.

יצרני ציוד היקפי (peripheral equipment)

חברות אלה מייצרות ציוד היקפי התואם לציוד של יצרני המחשבים. התאימות מתייחסת לאפשרות החלפה בין הציוד ההיקפי של יצרן המחשב לבין ציוד היקפי שמייצר יצרן אחר. הכניסה לתחום זה קלה משום מחסום הון נמוך, התמחות בסוג ציוד מסוים, שוק פוטנציאלי גדול ושינויים טכנולוגיים מעטים יחסית.

יצרני/משווקי תוכנה (בתי תוכנה software houses)

בתי תוכנה עוסקים בפיתוח תוכנת מערכות ומערכות תוכנה יישומית ובשיווק מוצרי תוכנה אחידים. תחום זה גדל מאוד ועתיד לגדול בקצב מהיר ביותר. הכניסה לתחום זה קלה מאחר שאין מחסומי הון, וקיים שוק פוטנציאלי גדול מאוד.

פיתוח מערכות תוכנה (software development) נעשה לפי דרישות הלקוח. החל משלב ניתוח המערכות וכלה בשלב היישום — "נתפרת" המערכת לפי צורכי הלקוח. לעומת זאת מוצרי תוכנה אחידים (חבילות תוכנה, software packages) הם מערכות תוכנה המפותחות על־ידי בית תוכנה במטרה לענות על דרישות כלליות ביישום מסוים, והן משווקות ללקוחות "כפי שהן".

בדרך־כלל מוצרי תוכנה אחידים עונים על רוב צורכי הלקוח. קיים הבדל ניכר במחיר בין מערכות תוכנה הנבנות לפי דרישת הלקוח לבין מערכות תוכנה אחידות.

ארגונים רבים נוכחו לדעת שעלות רכישת תוכנה נמוכה בדרך־כלל מעלות פיתוח עצמי, התוכנה טובה יותר, מתועדת היטב, ולעתים מסופקת עם אחזקה שוטפת.

יצרני/משווקי מערכות ממוחשבות (בתי מערכות, systems houses)

בתי מערכות משווקים מערכות ממוחשבות שלמות (turnkey systems) הכוללות חומרה ותוכנה, והעונות לדרישות יישום מסוים או לקבוצת יישומים. מוצרי התוכנה הם אחידים ומותאמים לציוד. תחום זה נמצא בגידול ומספר החברות בו הולך וגדל בהתמדה. קיים מחסום הון בכניסה לתחום זה, הנובע מעלות הפיתוח של מערכות התוכנה האחידות.

לשכות שירות (service bureaus)
לשכות השירות מחזיקות מערכות ממוחשבות הכוללות תוכנה במגוון רחב של יישומים. הן משווקות את שירותי המערכות הממוחשבות ללקוחותיהן. הגידול בתחום זה הוגבל לאחרונה, עקב הירידה היחסית במחירי המערכות הממוחשבות, שאיפשרה ללקוח הפוטנציאלי של לשכת השירות לרכוש מחשב המתאים לו, ולבצע את העיבודים בתוך הארגון.
הקשר בין הלקוח ללשכת השירות נעשה באחת משתי השיטות העיקריות:
— העברת תנועות לביצוע עיבוד בלשכת השירות, ובתום העיבוד העברת הפלטים ללקוח. שיטה זו (מכלול — batch) מקובלת ביישומים הדורשים עיבודים תקופתיים, כגון משכורת, הנהלת חשבונות.
— העמדת תחנת עבודה (כגון מסוף, מדפסת) במשרדי הלקוח, ושימוש בשירותי המערכת הממוחשבת הנמצאת בלשכת השירות באמצעות קווי תקשורת. שיטה זו (שיתוף זמנים — time sharing) מקובלת ביישומים הדורשים אינטראקציה רבה בין האדם והמערכת הממוחשבת, כגון תיכון באמצעות מחשב.

חברות ייעוץ (consulting firms)
חברות הייעוץ הן בעלות רקע רחב וידע רב בתחומי התמחותן. החברות יועצות ללקוחותיהן באפיון המערכת הממוחשבת המתאימה, בבחירתה (חומרה ותוכנה) ובשיטות היישום של המערכת הממוחשבת בארגון הלקוח. הכניסה לתחום זה קלה ביותר.

מגמות בתחום מערכות מידע ממוחשבות

התפתחותם המהירה וחדירתם של המחשבים לכל תחומי חיינו הנה ללא תקדים; מאז הופעתו של המחשב הראשון בשנת 1944, התקדמה טכנולוגיית המחשבים בקצב מדהים, יכולת הביצוע של המחשבים גדלה בעשרות מונים, ומחירם ירד בשיעורים גדולים. המחשב חודר היום לבתים, וזאת לאחר שחדר למרבית הארגונים, גדולים כקטנים.
מאז שנות השישים, יכולת הביצוע של המחשב ביחידת זמן גדלה פי 30, בעוד שעלות הביצוע קטנה פי 40! בשנת 1955 עלות ביצוע 10,000 פעולות אריתמטיות במחשב היתה כ־1.25 $. בשנת 1980 עלות הביצוע של אותן הפעולות היתה כ־0.01 $. להשוואה, התקדמות דומה בענפים אחרים היתה מביאה לכך שבשנת 1980 ניתן היה לייצר מכונית בכ־200 $, ולטוס מסביב לעולם בכ־3 $.
ההתפתחות המהירה והשיפורים הטכנולוגיים המיושמים בקצב מהיר מצביעים על המשך הנטייה להגדלת הביצועים ולהוזלת מחירי הציוד גם בעתיד. שיעור ההוצאה לחומרה קטן ביחס לסך כל התקציב לעיבוד נתונים של הארגון;

לעומת זאת שיעור ההוצאה לתוכנה הולך וגדל. ההדגשים של הארגון יעברו מהחומרה לתוכנה. מעורבות ההנהלה והמשתמשים בפיתוח וביישום מערכות ממוחשבות תגדל. ההתייחסות אל מערכות ממוחשבות כאל מערכות עיבוד נתונים תפנה את מקומה להתייחסות למערכות אלה כאל מערכות מידע. הנהלות יכירו בחשיבותו ובערכו של המידע כמשאב ארגוני, בעל חשיבות שאיננה פחותה מחשיבות משאבים אחרים, כגון הון, כוח־אדם, ציוד.
ננתח את המגמות בעתיד הקרוב בארבעה מישורים ראשיים: חומרה, תוכנה, יישומים, כוח־אדם, וכן את התוצאות שתצמחנה מן השימוש במערכות מידע ממוחשבות בארגון.

מגמות בתחום החומרה

התפתחויות טכנולוגיות מצביעות על המשך הנטייה לשפר את היחס מחיר/ביצועים של ציוד מערכות מחשב — המחשב עצמו, יחידות אחסון הנתונים ואמצעי הקלט והפלט. המהירות, האמינות, הקיבולת ופשטות השימוש ישופרו במקביל להוזלת המחיר.
חלק הולך וגדל מתוכנת המערכות ישולב בציוד כחלק אינטגרלי ממנו, וזאת כתוצאה ממיזעור ומהוזלת מחירי הזיכרון הראשי. מערכות הפעלה, בקרי תקשורת, בקרי מסד־נתונים, בקרת עיבודים ומעקב העבודות ישולבו בחומרה.
מגוון המסופים המשמשים ליישומים מיוחדים (כגון נקודות מכירה, כספומט) יגדל, ביצועיהם ישופרו ומחירם יוזל. מסופים אלה יהיו למעשה מחשבים זעירים — בני־תכנות ובעלי זיכרון פנימי — כך שחלק מהעיבוד יבוצע במסוף עצמו ללא צורך בהתקשרות למחשב המרכזי.
שיפור האמינות, הקיבולת והמהירות של אמצעי התקשורת, במקביל להוזלת מחירם, ישפיעו באופן ניכר על הרחבת השימוש ברשתות תקשורת מסחריות ובתקשורת באמצעות לוויינים.

מגמות בתחום התוכנה

לעומת הוזלת מחירי החומרה, גדל שיעור ההוצאה לתוכנה יחסית לסך־כל ההוצאה למערכות מידע ממוחשבות. השיפורים הטכנולוגיים בתחום התוכנה ופיתוח התוכנה מעטים יחסית לקצב פיתוח החומרה; רמת הדרישות של המשתמשים בתוכנה גדלה, ומחירם של אנשי מקצוע בתחום התוכנה עולה. הנהלות ארגונים יפנו תשומת־לב רבה יותר למחירי פיתוח, רכישה ואחזקה של תוכנה.
תוכנת מערכות (system software) — הכוללת תכניות כגון מערכת הפעלה, בקרי תקשורת, מערכות ניהול מסדי נתונים, תכניות שירות ומתרגמים, תשולב בציוד ותיהפך לחלק מהחומרה, דבר שיוזיל את מחירה ואת עלות אחזקתה.
תוכנת המערכות תשופר ותוכל לטפל בכמויות גדולות של נתונים, בסביבה הידברותית־מקוונת.

שפות התכנות יהיו טכניות פחות ו"טבעיות" יותר, דבר שיאפשר למשתמשים רבים לתכנת בעצמם ולהשתמש במערכת המחשב ללא תלות באנשי מקצוע בתחום התוכנה.

יתרחב השימוש במערכות לניהול מסדי נתונים שיתמכו בהזנת נתונים, עדכון ושליפת מידע בצורה מקוונת, והללו יסייעו בפיתוח מערכות יישומיות. מערכות אלה יכילו בקרי־מסד הנתונים ואמצעים לשמירה על פרטיות ועל בטיחות שיהיו חלק מהמערכת; פעילויות שבוצעו בעבר בצורה ידנית יבוצעו על־ידי התוכנה.

יפותחו ויאומצו טכניקות מתקדמות של ניתוח מערכות, של פיתוח מערכות תוכנה ושל תיעוד מערכות.

מגוון התוכנה היישומית — פיתוח עצמי של הארגון ובעיקר מוצרי תוכנה אחידים — יתרחב במידה רבה. התוכנה היישומית תהיה ברובה הידברותית־מקוונת. יותר ויותר יישומים יאספו את הנתונים במקור היווצרם, בזמן אמתי, ובאמצעים קריאים ישירות על־ידי המחשב. הפלט המודפס על נייר יצומצם במידה רבה.

מגמות בתחום כוח־אדם

עלות כוח האדם המקצועי היא מרכיב העלות הגדל בקצב המהיר ביותר מבין כל מרכיבי עלות מערכות המידע בארגון. נטייה זו תימשך בעתיד, והביקוש לכוח אדם מקצועי בתחום מערכות המידע הממוחשבות ילך ויגדל.

שלוש דרכים להקטנת עלות כוח האדם המקצועי:

- הסבת פעולות הנעשות באופן ידני לפעולות ממוחשבות
- הגדלת תפוקת כוח האדם המקצועי
- הגדלת תרומת המשתמש לעבודת הפיתוח.

מאמצים רבים מושקעים במטרה להסב פעולות הנעשות באופן ידני לפעולות ממוחשבות: מערכות הפעלה ומערכות לניהול מסדי נתונים מבצעות חלק גדול והולך מהפעילויות שבוצעו בעבר על־ידי אנשים. מסופים המשמשים לקליטת נתונים מאפשרים קיצור המהלך של קליטת הנתונים ואמותם, וחיסכון בעבודה הידנית שהושקעה בכך בעבר. תקשורת בין מערכות ממוחשבות ביטלה את הצורך בהעברת נתונים בשיטות ידניות. שילוב תוכנת המערכות (מערכות ההפעלה) בתוך החומרה מקטין את הצורך באחזקה ובפיתוח על־ידי תכניתני מערכות.

שיפור התפוקה של כוח אדם מקצועי — מנתחי מערכות, מעצבי מערכות ותכניתנים — יושג באמצעות כלים ממוחשבים לפיתוח מערכות ממוחשבות, הקיימים והנמצאים בפיתוח, והמאפשרים חיסכון בזמן הנדרש לפיתוח ופישוט אחזקה עתידית של המערכות.

הרחבת השימוש בשיטות הידברותיות לתיכון ותכנות בפיתוח מערכות חדשות ואחזקת מערכות קיימות מביאה להגדלת היעילות והתפוקה. אימוץ טכניקות של

ניהול פרוייקטים יצמצם את החריגות בהשקעת משאבים, ואת הפגמים בשלבי הפיתוח השונים. דרך אחרת לחסכון בכוח אדם היא על־ידי שימוש במוצרי תוכנה אחידים (חבילות תוכנה), אשר השימוש בהן ילך ויתרחב.

הגדלת חלקם של המשתמשים בעבודות פיתוח, יישום ואחזקה של המערכות תאפשר להוריד מהעומס המוטל על צוותי הפיתוח, התפעול והאחזקה. ייעשה שימוש באמצעים שונים כדי להשיג זאת: מערכות הידברותיות ומסופים הנמצאים אצל המשתמשים למטרות הזנת נתונים, שאילתות, עדכון קבצים ותכנות. המשתמשים ירכשו יותר ויותר ידע בתחום מערכות המידע, דבר שיאפשר להם להגדיר בצורה הטובה ביותר את צורכי המידע שלהם. יפותחו שפות תכנות הדומות לאנגלית, דבר שיקל על המשתמשים הלא־מקצועיים בתחום המחשב לפתח בעצמם את המערכות הדרושות.

ההשפעות על הארגון

השפעתן של מגמות אלה על ארגונים, תגרום להרחבת השימוש במערכות ממוחשבות בכל תחומי התפעול והניהול.

הסביבה הארגונית תהיה תלויה במידה רבה בעיבוד הנתונים; שיעור העובדים בכוח העבודה התלוי בצורה ישירה או עקיפה במערכות ממוחשבות לצורך ביצוע עבודתם ילך ויגדל. שיעור זה יגיע לכ־70 אחוז בעשור הקרוב. עם חדירת המחשבים הגדולים, הבינוניים, המיני והמיקרו, תעלה נטיה לפילוג כושר המיחשוב לכל היחידות הארגוניות. הוזלת המחירים של אמצעי האחסון והמסופים תתרום להרחבת השימוש במסופים ולפיתוח מערכות הידברותיות־מקוונות. מספר המסופים הקשורים ישירות למחשב יגדל בשיעור של כ־30 אחוז בשנה. שיפור אמצעי התקשורת יאפשר תקשורת למרחקים, וכתוצאה מכך יביא לשימוש מיטבי במידע בארגונים גדולים.

יעשה שימוש במערכות תוכנה רבות יותר ובתחומי יישום חדשים ומתוחכמים יותר. הרחבת השימוש במסדי נתונים בסביבה הידברותית תעמיד כמויות גדולות של נתונים לרשות משתמשים רבים בארגון, עם אמצעי הגנה ואבטחה נאותים.

תהליכי הבקרה והביקורת של מערכות מידע ממוחשבות יקבלו משקל רב, מאחר שפעילויות עיבוד נתונים שנבדקו בעבר בצורה ידנית יבדקו במידה מרובה על־ידי תכניות אחרות.

יותר ויותר ארגונים יצטרפו לחוג המשתמשים בטכנולוגיית המידע, ומשתמשים ותיקים יעברו למערכות מתוחכמות יותר. אפשרות השימוש במיני ובמיקרו מחשבים, בשילוב השימוש במוצרי תוכנה אחידים, ידחוף ארגונים קטנים לעבור מעבודה ידנית למערכות ממוחשבות. ככל שמחיר עיבוד הנתונים ירד, יהפכו יישומים שלא היו מוצדקים מבחינת המחיר בעבר לכדאיים ביותר בעתיד.

יישומים הנעשים במכלול, יפנו מקומם ליישומים הידברותיים־מקוונים. מערכות מידע ריכוזיות יוסבו לרמה מסוימת של פעילות מפולגת. היישומים המשרתים

את הרמה התפעולית בארגון, ישופרו ליישומים התומכים בקבלת החלטות ברמת הניהול הבכיר.
התלות הרבה של הארגונים במערכות המידע הממוחשבות תחייב הפניית תשומת־לב רבה יותר של ההנהלה לנושא. המידע הנו משאב ארגוני חיוני, שיש לתכננו, לבקרו ולהשתמש בו, כך שתופק ממנו התועלת המירבית. תשומת־לב ההנהלה למשאבי המידע עשויה לעלות על תשומת־הלב הניתנת כיום למשאבים אחרים, כגון לכוח־אדם, להון, למכונות ולחמרים.

מגמות בתחום היישומים

יישומים עתידיים יקרבו מאוד את המחשב הן למשתמשים בתוך הארגון והן לציבור.
קיימים תחומים בולטים, שבהם ניכרת מגמה ברורה של השפעת המחשב:

— תיכון וייצור באמצעות מחשב (תיב״מ — CAD—CAM Computer Aided Design/Manufacturing).
— מערכות ממוחשבות שיסייעו בכל שלבי התיכון והייצור של מוצרים, החל משלב התרשימים הראשוניים וכלה בשלב הייצור המסחרי.
— מערכות תומכות החלטה (Decision Support Systems). מערכות עיבוד התנועות ישופרו למערכות שיספקו מידע מהיר ומקיף, שיסייע בתהליכי קבלת ההחלטות של הניהול הבכיר.
— המשרד הממוחשב (Office Automation).
מערכות ממוחשבות יבצעו פונקציות משרדיות, כגון עיבוד תמלילים, דואר אלקטרוני, תיוק ושליפת מסמכים, תקשורת משרדית, תזמון פגישות וכד׳.
— מחשבים אישיים (Personal Computing).
מחשבים קטנים, נידים וזולים יאפשרו ביצוע עיבודים אישיים בכל מקום, ויאפשרו התקשרות באמצעות רשתות תקשורת למחשבים אחרים.
— נקודת מכירה (Point of Sale — POS).
גם מחוץ למשרד ולבית יגיעו האדם והמחשב למגע קרוב. התחום הבולט הנו נקודות מכירה בחנויות שיפעלו כקופות רושמות ממוחשבות, כספומט, תשלומט וכד׳, שיבצעו העברת כספים אלקטרונית בין מחשבים, דבר שיגרום, בסופו של דבר, לחברה ללא מזומנים.

מערכות מידע ממוחשבות נועד למנהלים ולאנשי ארגונים החשופים לטכנולוגיות המידע המודרניות. מטרת הספר להקנות לקורא בסיס להבנת הפעולה, המבנה והשימושים של המחשבים ומערכות המידע, וכן להעמיד עבורו טרמינולוגיה שתאפשר לו תקשורת טובה עם מומחי מחשבים. מטרת הספר אינה ליצור מקוראיו מומחים למחשבים, מנתחי מערכות או תכניתנים.

מערכות מידע ממוחשבות בא לספק את הרקע העיוני ללימוד נושאי המחשב ומערכות המידע הממוחשבות ויישומם בארגון. הספר מבוסס על נסיון ההוראה של המחברים בנושאים אלה בקורסים לתלמידי מנהל עסקים ולתלמידי התכניות השונות להכשרה ולקידום מנהלים.

מערכות מידע ממוחשבות כולל את הפרקים:

• מערכות מידע - הגדרה, תכונות ומבנה • מערכת המחשב - חומרה, תוכנה • תכנות • טכנולוגיות עיבוד מידע • יישום מערכות מידע ממוחשבות • היבטים ארגוניים וניהוליים של מערכות מידע ממוחשבות • ענף המחשבים ומגמות לעתיד.

צ׳ריקובר מוציאים לאור בע״מ